FVL高分解能センサSDK/DOS(FVL-SDK-HRS/DOS)
FVL基本SDK/LNX(FVL-SDK-BSC/LNX)

902
高分解能センサ
ライブラリ説明書

☆第１３版☆

（株）ファースト

御注意

◎本書の内容の一部または全部を無断で転載することは固くお断りします。
◎本書の内容について、将来改良を目的に予告なしに変更することがあります。
◎Watcom C™は、Sybase Inc.とその関連会社の商標です。
◎Pentium™は、Intel Corporationの商標です。

はしがき

この説明書は、

ＦＶＬ基本ＳＤＫ／９０２(902画像処理ﾗｲﾌﾞﾗﾘ)　Ver.3.40

ＦＶＬ高分解能センサＳＤＫ／ＤＯＳ(902高分解能ｾﾝｻﾗｲﾌﾞﾗﾘ)　Ver.3.30

ＦＶＬ基本ＳＤＫ／ＬＮＸ　Ver.1.30

以降に対応しています。

本書は、９０２シリーズ用ユーザプログラム作成時に使用できる、「９０２　高分解能センサライブラリ」について記載したものです。

なお、この他に９０Ｘシリーズライブラリとしては、下記のようなものがあります。

◎９０Ｘ　基本ライブラリ（Ｖｏｌ．１，Ｖｏｌ．２，Ｖｏｌ．３）説明書

◎９０Ｘ　２値画像ライブラリ説明書

◎９０Ｘ　濃淡画像ライブラリ説明書

◎９０Ｘ　ビジョン・ツール・ライブラリ説明書

◎９０Ｘ　キャリパーライブラリ説明書

ユーザプログラムの開発方法その他につきましては

◎ＦＨＣ３３０Ａ取扱説明書

◎ＦＨＣ３３１ＬＶ取扱説明書

◎ＦＨＣ３３２１取扱説明書

◎ＦＨＣ３３１０(Ａ)取扱説明書

◎９０Ｘ　操作説明書

◎FAST　Vision　Libraryプログラマーズガイド

◎ＣＳＣ９０２ＳＴ取扱説明書

◎ＦＶ９０２取扱説明書

◎ＦＶＬ／ＬＮＸ操作説明書

をご参照ください。

目　　次

1．概　　要 ····· 1

2．付属ソフトウェア ····· 5

3．キャプチャボード ····· 9

3.1　搭載ボードの確認 ····· 11

3.2　２枚実装 ····· 12

3.3　画像にノイズが発生する場合 ····· 13

4．ユーザプログラム開発手順 ····· 15

4.1　ライブラリのインストール ····· 17

4.2　ユーザプログラムの開発手順 ····· 19

5．ラインセンサ・カメラ制御ライブラリ ····· 21

5.1　サンプルプログラム ····· 24

ラインセンサ・カメラ処理の初期化（Lib_lsensor_init） ---------- 27

ラインセンサ・カメラ処理の終了（Lib_lsensor_finish） ---------- 32

ラインセンサ画像の入力開始（Lib_lsensor_video_input） ---------- 33

画像の入力開始Ⅱ（Lib_lsensor_xvideo_input） ---------- 35

ラインセンサ画像取り込み完了ステータスの取得（Lib_lsensor_get_video_status） -------- 37

ラインセンサ画像入力のフリーズ（Lib_lsensor_freeze） ---------- 38

ラインセンサ調整画像入力（Lib_lsensor_test_input） ---------- 39

ビデオ入力チャネル設定(Lib_lsensor_set_video_channel)---------- 41

ビデオ入力チャネル取得(Lib_lsensor_get_video_channel)---------- 43

ビデオ入力チャネル数の取得(Lib_lsensor_get_video_channel_num)---------- 44

エラー情報の取得（Lib_lsensor_get_error_inf） ---------- 45

濃淡メモリの確保（Lib_lsensor_alloc_gray_memory） ---------- 46

２値メモリの確保（Lib_lsensor_alloc_bin_memory） ---------- 47

濃淡メモリのベース・アドレス通知（Lib_lsensor_adrs_gray_memory） ---------- 48

２値メモリのベース・アドレス通知（Lib_lsensor_adrs_bin_memory） ---------- 49

濃淡メモリの解放（Lib_lsensor_free_gray_memory） ---------- 50

２値メモリの解放（Lib_lsensor_free_bin_memory） ---------- 51

濃淡カレントメモリＮｏの参照（Lib_lsensor_get_gray_memory） ---------- 52

２値カレントメモリＮｏの参照（Lib_lsensor_get_bin_memory） ---------- 53

濃淡＆２値メモリの切り換え（Lib_lsensor_change_memory） ---------- 54

濃淡メモリの切り換え（Lib_lsensor_change_gray_memory） ---------- 55

２値メモリの切り換え（Lib_lsensor_change_bin_memory） ---------- 56

２値メモリクリア（Lib_lsensor_bin_memory_cls） ---------- 57

濃淡メモリクリア（Lib_lsensor_gray_memory_cls） ---------- 58

フレーム・バッファへの画像転送（Lib_lsensor_video_transmit） ---------- 59

表示ウインドウ範囲の取得（Lib_lsensor_get_extract_window） ---------- 60

表示ウインドウの設定（Lib_lsensor_set_extract_window） ---------- 61

縮小画像からの画像選択表示（Lib_lsensor_extraction_display_memory） ---------- 63

２値レベルの調整（Lib_lsensor_adjust_bin_level） ---------- 65

画像入力メモリのリセット（Lib_lsensor_memory_reset） ---------- 67

取り込みモードの設定(Lib_lsensor_SetGrabMode)---------- 68

目　　次

諸設定(Lib_lsensor_SetGrabCondition) ---- 71
エンコーダ入力の設定(Lib_lsensor_SetEncoder) ---- 73
キャプチャカードハードウェアバージョン取得（Lib_lsensor_hardver） ---- 76
取込中断・停止処（Lib_lsensor_video_input_abort） ---- 77
ラインセンサ連続取込開始（Lib_lsensor_continuous_grab） ---- 78
ラインセンサ取込確定枚数取得（Lib_lsensor_get_grab_count） ---- 79
ラインセンサ取込ライン数取得（Lib_lsensor_get_line_count） ---- 80
ラインセンサ用キャプチャパラメータの生成(Lib_lsensor_create_param) ---- 81
キャプチャパラメータの破棄(Lib_lsensor_destroy_param) ---- 85
キャプチャパラメータの設定(Lib_lsensor_set_param) ---- 86
キャプチャパラメータの取得(Lib_lsensor_get_param) ---- 87

6．高分解能エリアセンサ制御ライブラリ ・・・・・・・・・・・・・・・・・・・・ 91
6.1　本ライブラリをご使用の前に ・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 94
6.2　カメラ側の設定 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 98
高分解能エリアセンサ画像取り込み処理の初期化(Lib_high_res_init) ---- 106
高分解能画像取り込み処理の初期化Ⅱ(Lib_high_res_xinit) ---- 110
高分解能エリアセンサ画像取り込み処理の終了(Lib_high_res_finish) ---- 115
高分解能エリアセンサ画像の入力（フリーズ）(Lib_high_res_freeze) ---- 116
高分解能エリアセンサ画像の入力開始(Lib_high_res_xfreeze) ---- 117
高分解能エリアセンサ画像取り込み完了ステータスの取得(Lib_high_res_get_video_status) 118
高分解能エリアセンサ画像取り込み強制終了(Lib_high_res_stop) ---- 119
ビデオ入力チャネル設定(Lib_high_res_set_video_channel) ---- 120
ビデオ入力チャネル取得(Lib_high_res_get_video_channel) ---- 121
諸設定(Lib_high_res_SetGrabCondition) ---- 122
ビデオ入力チャネル数の取得(Lib_high_res_get_video_channel_num) ---- 124
縦方向２画素１６ビット出力カメラ画像の変換(Lib_high_res_convert_for_bit16b) ---- 125
高精細エリアカメラ用キャプチャパラメータの生成(Lib_high_res_create_param) ---- 127
キャプチャパラメータの破棄(Lib_high_res_destroy_param) ---- 132
キャプチャパラメータの設定(Lib_high_res_set_param) ---- 133
キャプチャパラメータの取得(Lib_high_res_get_param) ---- 134
6.3　サンプルソフトウェア説明 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 142

7．高分解能モード移行ライブラリ ・・・・・・・・・・・・・・・・・・・・・・・・ 150
高分解能モードの初期化(Lib_ExtraCameraInit) ---- 153
高分解能モードの終了(Lib_ExtraCameraFinish) ---- 154
7.1　処理の流れ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 155
7.2　「IPM-8500」及び「902」の初期化 ・・・・・・・・・・・・・・・・・・・・・・・ 156
7.3　画像取り込み ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 160
7.4　高分解能センサ画像の取り込み終了 ・・・・・・・・・・・・・・・・・・・・・・・ 162

1. 概　　要

本書は９０２シリーズおよびＦＶＬ／ＬＮＸシリーズの高分解能センサライブラリについて記載したもので、以下のようなライブラリがあります。

① ライセンサ・カメラ制御ライブラリ

② 高分解能エリアセンサ制御ライブラリ

③ 高分解能モード移行ライブラリ

注１　ＦＶＬ／ＬＮＸでは「高分解能モード移行ライブラリ」は使用できません。
注２　ＦＶＬ／ＬＮＸでは９０Ｘ基本ライブラリ説明書（Vol.1）に記載されている、
「ビデオ入力制御ライブラリ」で高分解能エリアセンサをサポートしています。
新規に高分解能エリアセンサを用いたプログラムを作成する場合は、
「ビデオ入力制御ライブラリ」をお使い下さい。

なお、各ライブラリ関数の表記

９０１	９０２	９０３	９０４	FVL/LNX

は、９０１，９０２，９０３，９０４，ＦＶＬ／ＬＮＸ使用時にライブラリ関数が使用できるかどうかを表すもので

○ ‥‥‥‥‥ 使用可能
空白‥‥‥‥‥ 使用不可

となっております。

2. 付属ソフトウェア

ライブラリ、サンプルソフトウェアは３．５インチフロッピーディスク（１．４４ＭＢフォーマット）またはＣＤ－ＲＯＭ“FAST Software Pack”にて提供されます。

①フロッピーディスクの内容

ＩＮＳＴ９０１．ＥＸＥ……インストールプログラム本体
ＩＮＳＴＡＬＬ．ＢＡＴ……インストールバッチファイル
ＲＥＡＤＭＥ．ＨＲ２……ライブラリ、サンプルソフト最新情報
ＬＩＢ¥＊．ＬＩＢ……ライブラリ
ＩＮＣＬＵＤＥ¥＊．Ｈ……ライブラリ・インクルードファイル
ＳＡＭＰＬＥ¥＊．Ｃ……サンプルソフトＣソース一式
ＥＸＥ¥＊．ＥＸＥ……サンプル実行ファイル一式

なお、フロッピーディスクの内容については、予告無く変更されることがあります。
フロッピーディスク内のＲＥＡＤＭＥ．ＨＲ２ファイルにて最新の情報を提供しますので、必ず、お読みください。

②ＣＤ－ＲＯＭの内容

（９０２シリーズの場合）
Q:¥software¥902hr　に以下のファイルが含まれています。

ＳＥＴＵＰ．ＥＸＥ ……セットアッププログラム
ＳＥＴＵＰ．ＩＮＩ
ＳＥＴＵＰ．ＩＮＳ
ＳＥＴＵＰ．ＰＫＧ
ＳＥＴＵＰ．ＤＢＧ
ＩＳＤＢＧＮ．ＤＬＬ
ＤＩＳＫ１．ＩＤ
ＤＡＴＡ１．Ｚ
＿ＳＥＴＵＰ．ＬＩＢ
＿ＳＥＴＵＰ．ＤＬＬ
＿ＩＳＤＥＬ．ＥＸＥ
＿ＩＮＳＴ３２Ｉ．ＥＸ＿

（ＦＶＬ／ＬＮＸシリーズの場合）
Q:¥software¥FVL_LNX¥LIB　に以下のファイルが含まれています。

ＳＥＴＵＰ．ＥＸＥ ……セットアッププログラム
ＳＥＴＵＰ．ＩＮＸ
ＳＥＴＵＰ．ＩＮＩ
ＳＥＴＵＰ．ＢＭＰ
ＬＡＹＯＵＴ．ＢＩＮ
ＩＫＥＲＮＥＬ．ＥＸ＿
ＤＡＴＡ１．ＨＤＲ
ＤＡＴＡ１．ＣＡＢ
ＤＡＴＡ２．ＣＡＢ

※Q は CD-ROM ドライブを意味します。お使いの PC により異なります。

3. キャプチャボード

3.1　搭載ボードの確認

キャプチャボードは、製品により搭載されるボードが異なります。
本ライブラリは以下のキャプチャボードに対応しています。

（９０２シリーズの場合）

ボード型式	ラインセンサ	高分解能エリアセンサ	メーカ
ＩＰＭ－８５００Ａ	○		（株）グラフイン
ＩＰＭ－８５００Ｄ		○	（株）グラフイン
ＦＨＣ３３０Ａ（推奨）	○		（株）ファースト
ＦＨＣ３３１（推奨）		○	（株）ファースト
ＦＨＣ３３１ＬＶ（推奨）		○	（株）ファースト
ＦＨＣ３３２１（推奨）		○	（株）ファースト
ＦＨＣ３３１０(Ａ)（推奨）	○	○	（株）ファースト

（ＦＶＬ／ＬＮＸの場合）

ボード型式	ラインセンサ	高分解能エリアセンサ	メーカ
ＦＨＣ３３１		○	（株）ファースト
ＦＨＣ３３１ＬＶ		○	（株）ファースト
ＦＨＣ３３２１		○	（株）ファースト
ＦＨＣ３３１０(Ａ)	○	○	（株）ファースト

3.2　２枚実装

本ライブラリは以下の組み合わせで、キャプチャボードを２枚までサポートします。

- ＩＰＭ－８５００Ａ　＋　ＩＰＭ－８５００Ａ
- ＩＰＭ－８５００Ｄ　＋　ＩＰＭ－８５００Ｄ
- ＩＰＭ－８５００Ａ　＋　ＩＰＭ－８５００Ｄ
- ＦＨＣ３３０Ａ　＋　ＦＨＣ３３０Ａ　（推奨）
- ＦＨＣ３３１（ＬＶ）＋　ＦＨＣ３３１（ＬＶ）（推奨）
- ＦＨＣ３３２１　＋　ＦＨＣ３３２１　（推奨）
- ＦＨＣ３３１０(Ａ)　＋　ＦＨＣ３３１０(Ａ)　（推奨）

ビデオ入力対象ボードの切り替えは、チャネル設定ライブラリで行います。
チャネルとキャプチャボードの対応については
　ＦＨＣ３３０Ａ取扱説明書
　ＦＨＣ３３１ＬＶ取扱説明書
　ＦＨＣ３３２１取扱説明書
　ＦＨＣ３３１０(Ａ)取扱説明書
を参照してください。
ＩＰＭ８５００Ａ，Ｄについては、ＰＣＩバス認識順序に従います。

3.3　画像にノイズが発生する場合

画像取り込みを行なった際に、カメラの個体差、ケーブル長の差などの原因により取り込み画像に点状のノイズが現れる場合があります。
画像にノイズが発生する場合はサンプリングクロックの遅延設定、またはサンプリングクロックの位相設定によって、画質が改善する可能性があります。

サンプリングクロックの遅延設定を変更する場合は、各キャプチャーボードの取扱説明書を参照してください。

サンプリングクロックの位相設定はＦＨＣ３３１ＬＶを除きソフトウェアで行ないます。詳細は後述のライブラリ説明を参照してください。
ＦＨＣ３３１ＬＶをお使いの場合は、ボード上のジャンパで位相設定を行ないます。
詳細は「ＦＨＣ３３１ＬＶ取扱説明書」をご覧ください。

4. ユーザプログラム開発手順

4.1　ライブラリのインストール

インストールには“高分解能センサライブラリ・サンプルソフトＦＤ”またはＣＤ－ＲＯＭ“FAST Software Pack”が必要です。
キャプチャボードにＩＰＭ－８５００Ａ，ＩＰＭ－８５００Ｄを使用する場合は、キャプチャボードに付属の“ＩＰＭ－８５００制御ライブラリ（ＤＯＳ　Ｌｉｂ）ＦＤ”が必要です。

①フロッピーディスクからのインストール

○高分解能センサライブラリのインストール

高分解能センサライブラリ・サンプルソフトＦＤをフロッピードライブに挿入し、カレントドライブをフロッピードライブに変更します。

A>INSTALL　転送元ドライブ　転送先ドライブ　（ディレクトリ）

と入力します。
インストールプログラムが立ち上がりますので、コンソールに表示されるメッセージに従って操作してください。
インストールするディレクトリは省略することができます。
省略するとディレクトリ“ＣＳＣ９０２”の下にインストールします。
操作が終了すると以下のようなディレクトリ構造となります。

＜ＣＳＣ９０２＞ ‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ ユーザソフト開発用カレントディレクトリ
＜ＣＳＣ９０２＞＜ＬＩＢ＞ ‥‥‥‥‥‥‥‥‥‥‥‥ ライブラリ
＜ＣＳＣ９０２＞＜ＩＮＣＬＵＤＥ＞ ‥‥‥‥‥‥‥ インクルードファイル
＜ＣＳＣ９０２＞＜ＳＡＭＰＬＥ＞ ‥‥‥‥‥‥‥‥‥ サンプルプログラム

○ＩＰＭ－８５００制御ライブラリのインストール

インストールプログラムが用意されていませんので、フロッピーの中の
ＩＰＭ８５ＬＩＢ．ＬＩＢ
を高分解能センサライブラリと同じディレクトリにコピーしてください。
キャプチャボードにＦＨＣ３３０Ａ，ＦＨＣ３３１（ＬＶ），ＦＨＣ３３１０(Ａ)を使用する場合は必要ありません。

②ＣＤ－ＲＯＭからのインストール

○高分解能センサライブラリのインストール

Windows のスタートメニューより「ファイル名を指定して実行」を選択し、
Q:¥software¥902hr¥setup.exe　を実行してください。（９０２シリーズの場合）
Q:¥software¥FVL_LNX¥LIB¥setup.exe　を実行してください。（ＦＶＬ／ＬＮＸの場合）
※Q は CD-ROM ドライブを意味します。お使いの PC により異なります。

セットアッププログラムが起動しますので、表示されるメッセージに従いインストールを実行してください。
インストーラのデフォルトでインストールした場合、以下のディレクトリにインストールされます。

（９０２シリーズの場合）
＜ＣＳＣ９０２＞＜ＬＩＢ＞･･･････････････ライブラリ
＜ＣＳＣ９０２＞＜ＩＮＣＬＵＤＥ＞･･･････インクルードファイル
＜ＣＳＣ９０２＞＜ＳＡＭＰＬＥ＞･･･････････サンプルプログラム

（ＦＶＬ／ＬＮＸの場合）
＜ＦＶＬ／ＬＮＸ＞＜ＬＩＢ＞･･･････････････ライブラリ
＜ＦＶＬ／ＬＮＸ＞＜ＩＮＣＬＵＤＥ＞･･･････インクルードファイル
＜ＦＶＬ／ＬＮＸ＞＜ＳＡＭＰＬＥ＞･･･････サンプルプログラム

○ＩＰＭ－８５００制御ライブラリのインストール

CD-ROM には含まれていません。「①フロッピーディスクからのインストール」を参照し、インストールしてください。

4.2　ユーザプログラムの開発手順

ユーザプログラムの開発手順の詳細につきましては、『FAST Vision Library プログラマーズガイド』を参照してください。

ここでは、９０２シリーズでの本ライブラリのリンク方法について説明しますが、**使用するキャプチャボードによりリンクするライブラリが異なります**ので注意が必要です。

○ＷＬＮＫ９２．ＢＡＴを使用してリンクを行う場合

リンカオプションの入ったテキストファイルＣＳＣ９０２．ＣＦＧに高分解能センサライブラリを追加する必要があります。

［ＦＨＣ３３０Ａ，ＦＨＣ３３１（ＬＶ），ＦＨＣ３３２１］

※ＣＳＣ９０２．ＣＦＧへの追加例

```
ＬＩＢＲＡＲＹ　ｌｉｂ￥９０２ｓｙｓ．ｌｉｂ
　　，ｌｉｂ￥９０２ｂａｓｉｃ．ｌｉｂ
　　，ｌｉｂ￥９０２＿ＦＨＣ．ＬＩＢ　　← 追加
```

［ＦＨＣ３３１０(Ａ)］

※ＣＳＣ９０２．ＣＦＧへの追加例

```
ＬＩＢＲＡＲＹ　ｌｉｂ￥９０２ｓｙｓ．ｌｉｂ
　　，ｌｉｂ￥９０２ｂａｓｉｃ．ｌｉｂ
　　，ｌｉｂ￥ＦＨＣ３３１０．ＬＩＢ　　← 追加
```

［ＩＰＭ８５００Ａ，ＩＰＭ８５００Ｄ］

※ＣＳＣ９０２．ＣＦＧへの追加例

```
ＬＩＢＲＡＲＹ　ｌｉｂ￥９０２ｓｙｓ．ｌｉｂ
　　，ｌｉｂ￥９０２ｂａｓｉｃ．ｌｉｂ
　　，ｌｉｂ￥９０２＿ＨＲ２．ＬＩＢ　　← 追加
```

○ＷＡＴＣＯＭ統合開発環境を使用する場合

統合環境のＳｏｕｒｃｅ　ｆｉｌｅｓ：　に

［ＦＨＣ３３０Ａ，ＦＨＣ３３１（ＬＶ），ＦＨＣ３３２１］

９０２＿ＦＨＣ．ＬＩＢ

［ＦＨＣ３３１０(Ａ)］

ＦＨＣ３３１０．ＬＩＢ

［ＩＰＭ８５００Ａ，ＩＰＭ８５００Ｄ］

９０２＿ＨＲ２．ＬＩＢ、ＩＰＭ８５ＬＩＢ．ＬＩＢ

を追加してください。

９０２＿ＦＨＣ．ＬＩＢを使用する場合は、**９０２システムソフトVer.2.30以上が必要**です。

ＦＨＣ３３１０．ＬＩＢを使用する場合は、**９０２システムソフトVer.2.30以上が必要**です。

5. ラインセンサ・カメラ制御ライブラリ

本ライブラリはラインセンサ・カメラ・キャプチャボードを使用するためのライブラリです。
本ライブラリを使用するにあたっての留意事項を下記に記します。

ラインセンサ・キャプチャボードとして、ＩＰＭ－８５００Ａ，ＦＨＣ３３０Ａ，ＦＨＣ３３１０(Ａ)が必要です。

本ライブラリでサポートされているラインセンサ・カメラは ＩＰＭ－８５００Ａ、ＦＨＣ３３０Ａ、ＦＨＣ３３１０(Ａ) の仕様に準じます。詳細は、取扱説明書をご覧ください。
弊社で確認済みのラインセンサ・カメラは２００２年１０月現在、以下になります。

［ＩＰＭ８５００Ａ］

日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＹＨ２０４８Ｂ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＹＨ５０００Ｂ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＦＳ５１５０ＢＬ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＦＨ５０００Ｂ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＦＨ２０４８Ｂ
竹中システム機器株式会社	ＴＬ２０４８ＳＪＢ
竹中システム機器株式会社	ＴＬ５０００ＪＢ
株式会社エクセル	ＴＩ２０４８Ｍ

［ＦＨＣ３３０Ａ］

日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＹＨ５０００Ｂ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＦＨ５０００Ｂ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＮＹ５１５０ＢＬ

［ＦＨＣ３３１０(Ａ)］

三菱レイヨン株式会社	ＭＫＳシリーズ
日本エレクトロセンサリデバイス株式会社（ＮＥＤ）	ＮＵＦ７５００Ｄ
竹中システム機器株式会社	ＴＬ－７４５０ＵＦＤ
竹中システム機器株式会社	ＴＬ－５１５０ＵＦＤ
竹中システム機器株式会社	ＴＬ－２０４８ＦＤ

９０２シリーズのライブラリは、ＣＳＣ９０１シリーズのライブラリと比べて使用できなくなったライブラリと引数が変更になったライブラリがあります。

●使用できなくなったライブラリ
ランセンサ・カメラの初期化Ⅱ (Lib_lsensor_xinit)
ラインセンサ画像入力チャネル設定(Lib_lsensor_video_channel)
シェイディングＲＡＭ初期化(Lib_lsensor_init_shading_ram)
白／黒シェイディングＲＡＭの参照(Lib_lsensor_get_shading_ram)
白／黒シェイディングＲＡＭの更新(Lib_lsensor_set_shading_ram)
濃淡画像転送(Lib_lsensor_rs170_gmem_move)
２値画像転送(Lib_lsensor_rs170_bmem_move)

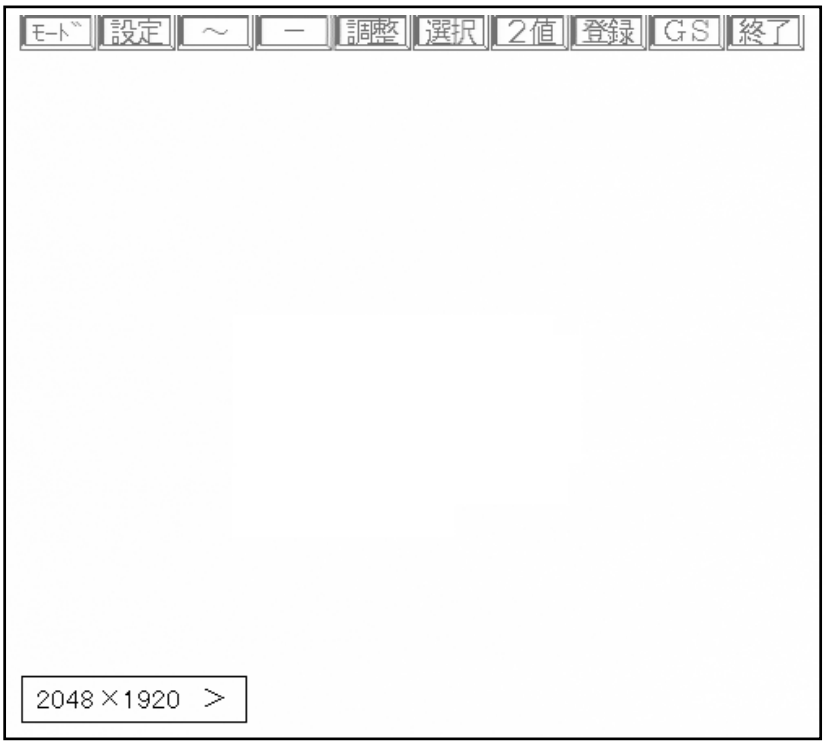
●引数が変更になったライブラリ
ラインセンサ・カメラ処理の初期化(Lib_lsensor_init)

5.1　サンプルプログラム

サンプルソースをコンパイル、リンクすると実行可能なサンプルプログラムとなります。
・ＩＰＭ－８５００Ａ．．．．．．．．．．．．．．．．lssmpl.c
・ＦＨＣ３３０Ａ・ＦＨＣ３３１０(Ａ)．．．．．．．．．l_sns_t.c
サンプルソースは、csc902¥sample ディレクトリにインストールされています。
以下はシステムの言語設定が日本語の場合の操作説明です。

■初期メニュー

サンプルプログラムを実行すると初期メニューが表示されます。



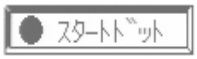
画面上部がメニューキーとなります。
画面下部の“２０４８×１９２０”は現在設定されている画像サイズを表します。
“＞”は現在設定されている表示モードを表し、
　　“＞”は縮小表示モード
　　“＝”は切り出し表示モード
です。

■モード

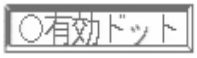
モードをクリックするごとに、縮小表示モードと切り出し表示モードが交互に切り替わります。
縮小表示モードで画像を取り込みと全体が縮小されて表示されます。
切り出し表示モードで画像を取り込むと表示ウィンドウ位置の画像が通常表示されます。

■設定

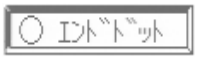
ここでラインセンサ・カメラの各種設定を行います。



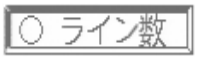
有効画素の前の無効画素数を設定します。



有効画素数を設定します。

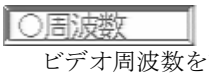


有効画素の後の無効画素数を設定します。



取り込むライン数を設定します。

クロックとビデオの比率を設定します。



ビデオ周波数を設定します。

■～

連続取り込みを開始します。
トラックボールのボタンを押すことによりメニューに戻ります。

■ー

ワンショット取り込みを行います。画面の右下には取り込み時間が表示されます。

■調整

ラインセンサ調整画像入力関数

Lib_lsensor_test_input()

を実行します。

使い方の詳細はライブラリ関数の説明を参照してください。

■選択

縮小画像からの画像選択表示関数

Lib_lsensor_extraction_display_memory()

を実行します。

使い方の詳細はライブラリ関数の説明を参照してください。

■２値

２値レベル調整関数

Lib_lsensor_adjust_bin_level()

を実行します。

使い方の詳細はライブラリ関数の説明を参照してください。

■登録

グレイサーチを行うためのサーチパタン登録

Lib_gs_exdefpat()

を実行します。

使い方の詳細は『９０Ｘ　濃淡画像ライブラリ説明書』を参照してください。

■ＧＳ

グレイサーチを実行し、パタンが見つかった場合はサーチ結果を表示します。

■終了

サンプルプログラムを終了します。

Lib_lsensor_init

機　能

ラインセンサ・カメラ処理の初期化

形　式

```
#include "f_lsens.h"
int Lib_lsensor_init( int video_freq, int start_dot, int scan_rate,
                      int h_start, int h_length, int v_length, int clock_video );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

映像入力をＲＳ１７０ビデオ入力からラインセンサ・カメラ入力に切り換え、初期化を行います。
各引数はラインセンサ・カメラの仕様に併せて設定してください。
また、**ラインセンサ・キャプチャボードＩＰＭ－８５００Ａ・ＦＨＣ３３０Ａ・ＦＨＣ３３１０(Ａ)で、引数の意味合いが異なります**ので、キャプチャボード毎に分けて説明します。

［ＩＰＭ－８５００Ａ］

① ***video_freq*** はビデオ周波数です。
クロックパルスビデオ比の設定する値によりビデオ周波数が変わります。
() 外は１：１、１：２の場合で () 内は２：１のビデオ周波数になります。

値	意　味
0	２０．０ＭＨｚ（１０．０　ＭＨｚ）
1	１３．３ＭＨｚ（　６．６５ＭＨｚ）
2	１０．０ＭＨｚ（　５．０　ＭＨｚ）
3	８．０ＭＨｚ（　４．０　ＭＨｚ）
4	６．７ＭＨｚ（　３．３５ＭＨｚ）
5	５．７ＭＨｚ（　２．８５ＭＨｚ）
6	５．０ＭＨｚ（　２．５　ＭＨｚ）
7	４．４ＭＨｚ（　２．２　ＭＨｚ）

② ***start_dot*** は映像出力開始位置です。
有効画素の前の無効画素数を指定してください。
設定範囲は１～５１２です。

③ ***scan_rate*** は蓄積時間（スキャンレート）です。
最小蓄積時間（１スキヤン当たりの最小必要時間）以上を画素数で指定します。
（最大２５０８８まで）

④ ***h_start*** は水平取り込み開始位置です。
０～［有効画素数－水平取り込み長］を画素数で指定します。
通常は０です。設定範囲は start_dot と h_start を加算して１～５１２です。

⑤ ***h_length*** は水平取り込み長です。
有効画素数以下を５１２～５１２０の範囲で指定します。
ここで指定する値は３２の倍数である必要があります。

⑥ ***v_length*** は垂直取り込み長（ライン数）です。
設定範囲は１～８１９１です。

⑦ ***clock_video*** はクロックパルスビデオ比です。
入力クロックパルスに対応して出力されるビデオ信号を指定します。

値　　定　　数	意　　味
0	１：１
1	１：２
3	２：１
4	外部入力クロック

外部入力クロックを使用する場合はクロックパルスビデオ比が１：１のカメラのみ使用可能です。

［ＦＨＣ３３０Ａ］

① ***video_freq*** はビデオ周波数です。
クロックパルスビデオ比の設定する値により設定可能なビデオ周波数が変わります。
() 内は設定可能なクロックパルスビデオ比です。

値	意　　味
2	２０．０ＭＨｚ（１：１、１：２）
3	１６．０ＭＨｚ（１：１、１：２）
4	１０．０ＭＨｚ（１：１、１：２）
5	８．０ＭＨｚ（１：１、１：２）
6	５．０ＭＨｚ（１：１）
7	４．０ＭＨｚ（１：１、１：２）
8	２．５ＭＨｚ（１：１）
9	２．０ＭＨｚ（１：１）

② ***start_dot*** は映像出力開始位置です。
有効画素の前の無効画素数を指定してください。設定範囲は０～４０９５です。

③ ***scan_rate*** は蓄積時間（スキャンレート）です。
最小蓄積時間（１スキヤン当たりの最小必要時間）以上を画素数で指定します。
（最大６５５３５まで）

④ ***h_start*** は水平取り込み開始位置です。
０～［有効画素数－水平取り込み長］を画素数で指定します。
通常は０です。設定範囲は start_dot と h_start を加算して０～４０９５です。

⑤ h_length は水平取り込み長です。
有効画素数以下を５１２～８１６０の範囲で指定します。ここで指定する値は３２の倍数である必要があります。また、４０９６以上指定する場合は、留意事項を必ず参照してください。

⑥ ***v_length*** は垂直取り込み長（ライン数）です。
設定範囲は１～１６０００です。４０９６以上指定する場合は、留意事項を必ず参照してください。

⑦ ***clock_video*** はクロックパルスビデオ比です。
入力クロックパルスに対応して出力されるビデオ信号を指定します。

値　　定　　数	意　　味
0	１：１
1	１：２

［ＦＨＣ３３１０(A)］

① ***video_freq*** はビデオ周波数です。
クロックパルスビデオ比の設定する値により設定可能なビデオ周波数が変わります。
() 内は設定可能なクロックパルスビデオ比です。

値	意　味
0	４０．０ＭＨｚ（１：１）
2	２０．０ＭＨｚ（１：１）
4	１０．０ＭＨｚ（１：１）
6	５．０ＭＨｚ（１：１）

② ***start_dot*** は映像出力開始位置です。
有効画素の前の無効画素数を指定してください。***h_start*** との合計で設定されます。
ＦＨＣ３３１０(A)のハードウェアバージョンによって設定範囲が変わります。

バージョン	設定範囲
0	０～２５５
3	０～１６３８３

ハードウェアバージョンは、Lib_lsensor_hardver関数で確認して下さい。

③ ***scan_rate*** は蓄積時間（スキャンレート）です。
最小蓄積時間（１スキャン当たりの最小必要時間）以上を画素数で指定します。
（最大６５５３５まで）

④ ***h_start*** は映像出力開始位置です。
start_dot との合計で設定されます。②を参照して下さい。

⑤ h_length は水平取り込み長です。
有効画素数を５１２～１６３６８の範囲で指定します。ここで指定する値は３２の倍数である必要があります。また、４０９６以上指定する場合は、留意事項を必ず参照してください。

⑥ ***v_length*** は垂直取り込み長（ライン数）です。
設定範囲は１～６５５３５です。４０９６以上指定する場合は、留意事項を必ず参照してください。

⑦ ***clock_video*** はクロックパルスビデオ比です。
入力クロックパルスに対応して出力されるビデオ信号を指定します。

値　定　数	意　味
0	１：１
1	１：２
2	１：４
3	２：１

戻り値

処理結果

値	定　　数	意　　味
０	NO_ERROR	正常終了しました。
－１	ARG_ERR	パラメータの値が異常な場合、またはメモリ不足です。
－１０	UNUSUAL_ERR	異常終了しました。
－９１	NO_DEVICE	このデバイスが存在しません。

例

ラインセンサ・カメラＦＨ２０４８Ｂ（ＮＥＤ）を使用します。

```
#include "f_lsens.h"

void line_sensor()
  {
    if( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
      {
        /*処理を行う*/
      }
    Lib_lsensor_finish();
  }
```

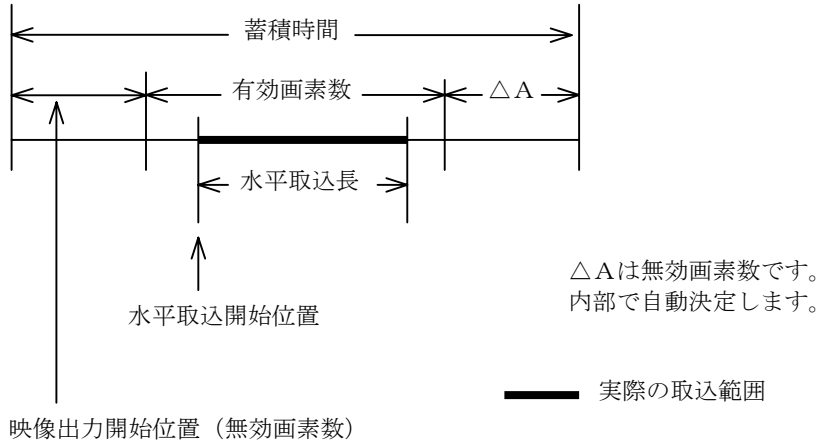
留意事項

○異常終了（-1）の場合は Lib_lsensor_get_error_inf でエラー情報を取得できます。
○ラインセンサ初期化時の必要メモリ容量
・濃淡画像メモリ　　＝ 水平取り込み長×垂直取り込み長
・２値画像メモリ　　＝ 水平取り込み長×垂直取り込み長÷８

例）２０４８×１９２０の場合
・濃淡画像メモリ　　２０４８×１９２０　　＝　３９３２１６０
・２値画像メモリ　　２０４８×１９２０÷８＝　　４９１５２０
合計　　約４．３Ｍｂｙｔｅ

濃淡画像メモリ、２値画像メモリは初期化時、必ず１面ごとに確保されます。画像処理ライブラリよっては２面必要な場合があります。
画像処理に必要なメモリサイズは、概ね以下の計算で求めることが可能です。
濃淡画像メモリ×ｎ＋２値画像メモリ×ｎ＋ワークメモリ
ｎ：必要画像メモリ数
ワークメモリ：濃淡画像メモリ÷２＋４Mバイト
＋グレイサーチパタンファイルサイズ（GRAY_PTN.SYS）
＋１次特徴量計測結果テーブルファイルサイズ（MEASURE.SYS）
○初期化後、終了する場合は必ず Lib_lsensor_finish 関数を実行してください。
○画像メモリサイズ（水平取り込み長×垂直取り込み長）が大きくなると一部画像処理ライブラリが正常に動作しない場合があります（概ね４０９６×４０９６を超える場合）。
・２値画像ライブラリ
１特徴量計測ライブラリの面積値、重心座標などの特徴量が正しく計測できない場合があります。この場合は、拡張１次特徴量計測ライブラリをご使用ください。
・濃淡画像ライブラリ
グレイサーチライブラリのサーチパタンサイズは、
Lb_gs_defadrs　を使用した場合は、最大１０２４×　５１２です。
Lib_gs_xdefadrs を使用した場合は、最大４０９６×４０９６です。
画像計測ライブラリの Lib_stddevi は４０９６×４０９６以上では、正しく計測できません。

○画像取り込み範囲は以下のようになります。



○画像取り込み時間は、以下の計算で概算を求めることが可能です。

ビデオ周波数×スキャンレート×垂直取り込み長

例）ビデオ周波数１０ＭＨｚ、スキャンレート５０００画素、垂直取り込み長４０００
０．１μｓ×５０００×４０００＝２０００００μｓ＝約２ｓ

○竹中システム機器株式会社のＴＬ－７４５０ＵＦＤ，ＴＬ－５１５０ＵＦＤ，
ＴＬ－２０４８ＦＤは、この関数では初期化できません。
後述の Lib_lsensor_set_param関数で初期化して下さい。

Lib_lsensor_finish

機　能

ラインセンサ・カメラ処理の終了

形　式

```
#include "f_lsens.h"
int Lib_lsensor_finish( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

ラインセンサ・カメラ入力の終了。

戻り値

処理結果

値	定　　数	意　　味
０	NO_ERROR	正常終了しました。
－１	ARG_ERR	異常終了しました。

例

ラインセンサ・カメラＦＨ２０４８Ｂ（ＮＥＤ）を使用します。

```
#include "f_lsens.h"

void line_sensor()
  {
    if( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
      {
        /*処理を行う*/
      }
    Lib_lsensor_finish();
  }
```

留意事項

○ Lib_lsensor_init を実行後、ユーザソフトを終了する場合は、必ずこの関数を実行してください。

Lib_lsensor_video_input

機　能

ラインセンサ画像の入力開始

形　式

```
#include "f_lsens.h"
int Lib_lsensor_video_input( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

最新のラインセンサ・カメラ入力画像をカレントメモリ上に入力します。
しかし、表示用フレーム・バッファには転送されませんので、画像をモニタＴＶに表示する場合には後述の Lib_lsensor_video_transmit を実行する必要があります。
本ライブラリは起動後、直ちに終了しますが、画像の入力は完了していません。
画像の入力完了は後述の Lib_lsensor_get_video_status で取得してください。

戻り値

処理結果

値	定　　数	意　　味
０	NO_ERROR	正常終了しました。
－１	ARG_ERR	パラメータの値が異常です。
－１０	UNUSUAL_ERR	異常終了しました。
－９１	NO_DEVICE	このデバイスが存在しません。

例

画像を入力後、フレーム・バッファに画像を転送します。

```
#include "f_lsens.h"

void input()
  {
    int status;

    if( NORMAL_RETURN == Lib_lsensor_video_input() )
      {
        for( status = ON; OFF != status ; )
          { /*起動後、取り込み完了まで、何か処理を行う。*/
            if( FINISHED_TRANSMIT == Lib_lsensor_get_video_status() )
              {
                Lib_lsensor_video_transmit( 0, GRAY_PLANE );
                status = OFF;
              }
          }
      }
  }
```

留意事項

○画像入力オフセットの変更はできません。
○ビデオ入力画像は濃淡画像のみです。
　２値画像を取得したい場合は“Lib_binary_convert”等により濃淡画像から２値画像への変換を行ってください。
○メモリ
　・濃淡画像はメモリ全体に取り込まれます。
○画像入力を起動中、ラインセンサ関係の関数を実行した場合の動作は保証しません。
○戻り値が UNUSUAL_ERR の場合、Lib_lsensor_get_error_inf により、異常内容を得る事ができます。

Lib_lsensor_xvideo_input

機　能

画像の入力開始Ⅱ

形　式

```
#include "f_lsens.h"
int Lib_lsensor_xvideo_input( memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

最新のラインセンサ・カメラ入力画像を指定されたメモリＮｏのメモリに入力します。
しかし、フレーム・バッファには転送されませんので、画像をモニタＴＶに表示する場合には後述の Lib_lsensor_video_transmit を実行する必要があります。
本ライブラリは起動後、直ちに終了しますが、画像の入力は完了していません。
画像の入力完了は後述の Lib_lsensor_get_video_status で取得してください。

① ***memory_no*** は入力先のメモリＮｏ．です。

戻り値

処理結果

値	定　　数	意　　味
０	NO_ERROR	正常終了しました。
－１	ARG_ERR	異常終了（パラメータの値が異常な場合）しました。
－１０	UNUSUAL_ERR	異常終了しました。
－９１	NO_DEVICE	異常終了（このデバイスが存在しない場合）しました。

例

メモリ１に画像を入力後、フレームバッファに画像を転送します。

```
#include "f_lsens.h"

#define  MEM_NO   1

void input()
{
  int status;

 if ( NORMAL_RETURN == Lib_lsensor_xvideo_input( MEM_NO ) )
  {
      for ( status = ON; OFF != status ; )
      { /* 起動後、取り込み完了まで、何か処理を行う。 */
          if ( FINISHED_TRANSMIT == Lib_lsensor_get_video_status() )
          {
              Lib_lsensor_memory_reset();
              Lib_lsensor_video_transmit( MEM_NO, GRAY_PLANE );
              status = OFF;
          }
      }
  }
}
```

留意事項

○画像入力オフセットの変更はできません。
○ビデオ入力画像は濃淡画像のみです。
　２値画像を取得したい場合は“Lib_binary_convert”等により濃淡画像から２値画像への変換を行ってください。
○メモリ
　・濃淡画像　メモリ全体に取り込まれます。
○画像入力を起動中、ラインセンサ関係の関数を実行した場合の動作は保証しません。
○戻り値が UNUSUAL_ERR の場合、Lib_lsensor_get_error_inf により、異常内容を得る事ができます。

Lib_lsensor_get_video_status

機　能

ラインセンサ画像取り込み完了ステータスの取得

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_video_status( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_video_input 実行後、画像取り込み完了ステータスを取得します。

戻り値

転送完了情報

値	定　　数	意　　味
0	FINISHED_TRANSMIT	画像入力が終了しました。
1	TRANSMITING	画像を入力中です。

例

画像を入力後、フレーム・バッファに画像を転送します。

```
#include "f_lsens.h"

void input()
  {
    int status;

    if( NORMAL_RETURN == Lib_lsensor_video_input() )
      {
        for( status = ON; OFF != status ; )
          {
            /*起動後、取り込み完了まで、何か処理を行う。*/
            if( FINISHED_TRANSMIT == Lib_lsensor_get_video_status() )
              {
                Lib_lsensor_video_transmit( 0, GRAY_PLANE );
                status = OFF;
              }
          }
      }
  }
```

留意事項

○画像入力を起動後、この関数でステータスを取得しないまま、画像メモリをアクセスした場合、メモリ内容の保証はできません。

Lib_lsensor_freeze

機　能

ラインセンサ画像入力のフリーズ

形　式

```
#include "f_lsens.h"
int Lib_lsensor_freeze( int transmit_flag );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

最新のラインセンサ・カメラ入力画像をカレントメモリ上に入力し、転送フラグの指定があればLib_lsensor_set_extract_windowで設定している画像表示ウインドウ位置をモニタＴＶ上に表示します。

① ***transmit_flag*** は転送フラグです。

値	定　　数	意　　味
1	TRANSMIT	カレントメモリ及び、フレーム・バッファに画像を転送します。
0	NOT_TRANSMIT	カレントメモリにのみ画像を転送します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

画像表示ウインドウを設定し、フリーズします。

```
#include "f_lsens.h"

void init()
  {
       Lib_init_cursor();
       if( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
         {
              Lib_lsensor_freeze( TRANSMIT );

              /*処理を実行*/
         }
  }
```

留意事項

○ Lib_lsensor_initt を実行前にコールする必要があります。
○メモリへの画像入力完了まで終了しません。
○ Lib_lsensor_freeze は転送フラグに TRANSMIT を指定された場合
Lib_set_extract_window で設定した画像表示ウィンドウ部分をフレーム･バッファに転送し、モニタＴＶに表示しています。

Lib_lsensor_test_input

機　能

ラインセンサ調整画像入力

形　式

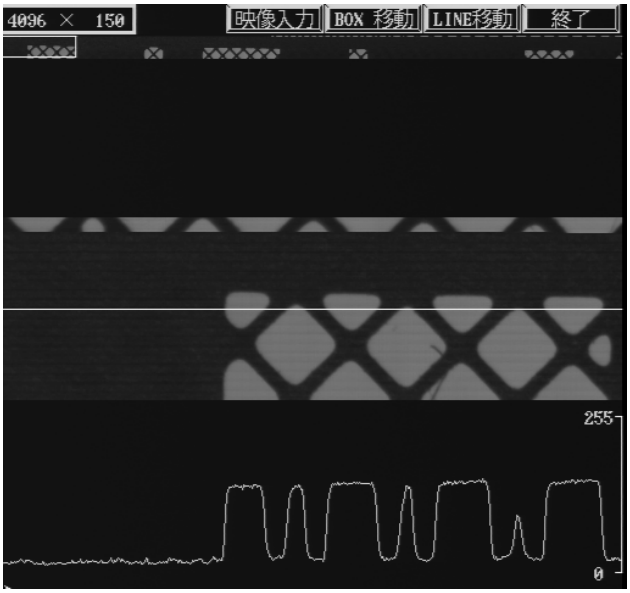
```
#include "f_lsens.h"
int Lib_lsensor_test_input( void  ( *user_process )( void ) );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

カメラの入力設定を行うために垂直取込長を１５０画素に固定し、入力を繰り返します。
上部に圧縮した全体画像を中央部にＢＯＸ内の切り出し画像を下部に切り出し画像の指定位置の濃度グラフを表示します。

① ***user_process*** は画像入力する関数へのポインタです。入力の際ＸＹステージなどの制御をする場合設定してください。ＮＵＬＬの場合は、内部で画像を入力します。



トラックボール操作

○映像入力(VIDEO In)
　映像を連続して入力します。入力を中止するときは実行キーを押してください。
○BOX 移動 (BOX Mov)
　切り出し表示用のウインドウを移動します。
　移動を中止するときは実行キーを押してください。
○LINE移動 (LINE Mov)
　濃度グラフを表示する部分を移動します。移動を中止するときは実行キーを押してください。
○終了 (Quit)
　調整画像入力を中止します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

初期化後、調整画像入力を実行します。

```
#include "f_lsens.h"

void init()
  {
       Lib_init_cursor();
       if( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
         {
              Lib_lsensor_test_input( NULL );
         }
  }
```

留意事項

○ Lib_lsensor_init を実行前にコールする必要があります。
○ Lib_init_cursor を実行前にコールする必要があります。
○画像入力用の関数はフレームバッファに画像を転送しないでください。
○ＦＨＣ３３０Ａの場合は、取り込みモードが“ノーマルモード”のみ使用可能です。“トリガモード”、“エンコーダモード”では使用できません。
○垂直取込長が 150 画素未満の場合は、調整入力の垂直取込長は画像メモリの垂直取込長と一致します。

Lib_lsensor_set_video_channel

機　能

ビデオ入力チャネル設定

形　式

```
#include "f_lsens.h"
void Lib_lsensor_set_video_channel( int channel );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

キャプチャボードを２枚実装されている場合には、本関数をコールすることで
カレントチャネル（ビデオ入力対象ボード）を切り替えることができます。

① ***channel*** はチャネル番号(０～１)です。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

ＣＨ１のラインセンサの初期化をします。

```
#include "f_lsens.h"

void set_video_channel()
  {
    if( NORMAL_RETURN == Lib_lsensor_set_video_channel( 1 ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
          {
            /*処理を行う*/
          }
        Lib_lsensor_finish();
      }
 }
```

留意事項

○本関数はキャプチャボードを２枚実装している場合のみ使用できます。
○本関数を呼び出さない場合のカレントチャネル番号は０です。
○以下の関数は、カレントチャネルに対して動作します。
Lib_lsensor_init,Lib_lsensor_finish,Lib_lsensor_video_input,
Lib_lsensor_get_video_status,Lib_lsensor_freeze,Lib_lsensor_test_input,
Lib_lsensor_xvideo_input
○画像入力中は、本関数でチャネルの変更は行わないでください。
○チャネル間で画像メモリのサイズが異なる場合は、必ず、ラインセンサ・カメラ処理の終了（Lib_lsensor_finish）後にチャネルを切り替えてください。
○ＩＰＭ－８５００ＡとＩＰＭ－８５００Ｄは共存が可能ですが、ＩＰＭ－８５００Ｄのチャネルに切り替える場合は、必ず、ラインセンサ・カメラ処理の終了（Lib_lsensor_finish）後にチャネルを切り替えてください。

Lib_lsensor_get_video_channel

機　能

ビデオ入力チャネル取得

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_video_channel( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

現在のカレントチャネル（ビデオ入力対象ボード）番号を取得します。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

キャプチャボードの実装枚数を取得し関数値として返します。

```
#include “f_lsens.h”

int get_channel( void )
{
        return( Lib_lsensor_get_video_channel() );
}
```

留意事項

ありません。

Lib_lsensor_get_video_channel_num

機　能

ビデオ入力チャネル数の取得

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_video_channel_num( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

ビデオ入力チャネル数（キャプチャボードの実装枚数）を取得します。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

ビデオ入力チャネル数を取得し関数値として返します。

```
#include “f_lsens.h”

int get_channel_num( void )
{
        return( Lib_lsensor_get_video_channel_num() );
}
```

留意事項

○ＩＰＭ－８５００ＡとＩＰＭ－８５００Ｄが搭載されている場合のチャネル数は２となります。

Lib_lsensor_get_error_inf

機　能

エラー情報の取得

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_error_inf( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

異常終了（ UNUSUAL_ERR ）が発生した場合のエラー情報を取得します。

戻り値

エラー情報を返します。

値	定　数	意　味
０	なし	正常終了しました。
－１	なし	パラメータ異常です。
－２	なし	メモリ不足です。
－９１	なし	ラインセンサカメラＩ／Ｆボードが存在しません。

例

画像入力時のエラー情報を取得し、表示します。

```
#include "f_lsens.h"

void input()
  {
    int  status;
    char string[30];

    if( NORMAL_RETURN == ( status = Lib_lsensor_video_input() ) )
        Lib_lsensor_video_transmit( 0, GRAY_PLANE );
    else if( UNUSUAL_ERR == status )
      {
        Lib_sprintf( string, "ERR:%d", Lib_lsensor_get_error_inf() );
        Lib_chrdisp( 1, 1, string );
      }
  }
```

留意事項

○ＦＨＣ３３１０(A)では、戻り値がＥＲＲＯＲ＿ＲＥＴＵＲＮ（－１）になります。

Lib_lsensor_alloc_gray_memory

機　能

濃淡メモリの確保

形　式

```
#include "f_lsens.h"
int Lib_lsensor_alloc_gray_memory( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

濃淡メモリ１面分のメモリ用ファイルを確保します。
なお、確保されたメモリ領域はランダムファイルとして生成されます。
ファイル名は「 GRAYMEM?.SYS 」（？＝メモリＮｏ）と自動的につけられます。

戻り値

確保したメモリＮｏを返します。

値	定　　数	意　　味
０以上	ありません。	確保したメモリＮｏを意味します。
－１	ERROR_RETURN	異常終了です。

例

メモリを新たに確保し、画像を入力し、何らかの処理を実行し、メモリを解放します。

```
#include "f_lsens.h"

void input()
  {
    int mem_no, cmem_no;
    char *adrs;

    cmem_no = Lib_lsensor_get_gray_memory(  );
    if( ERROR_RETURN != ( mem_no = Lib_lsensor_alloc_gray_memory() ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_change_gray_memory( mem_no ) )
          {
            if( NORMAL_RETURN == Lib_lsensor_video_input() )
              {
                Lib_lsensor_video_transmit( 1, GRAY_PLANE );
                adrs = Lib_lsensor_adrs_gray_memory( mem_no );
                /*何らかの処理*/
              }
          }
        Lib_lsensor_change_gray_memory( cmem_no );
        Lib_lsensor_free_gray_memory( mem_no );
      }
  }
```

留意事項

○確保されるメモリサイズ（バイト数）は
　　メモリサイズ ＝ 水平取込長 × 垂直取込長
○ラインセンサカメラ入力の初期化後は、Lib_alloc_gray_memory も同機能となります。

Lib_lsensor_alloc_bin_memory

機　能

２値メモリの確保

形　式

```
#include "f_lsens.h"
int Lib_lsensor_alloc_bin_memory( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

２値メモリ１面分のメモリ用ファイルを確保します。
なお、確保されたメモリ領域はランダムファイルとして生成されます。
ファイル名は「 BIN_MEM?.SYS 」（？＝メモリＮｏ）と自動的につけられます。

戻り値

確保したメモリＮｏを返します。

値	定　　数	意　　味
０以上	ありません。	確保したメモリＮｏを意味します。
－１	ERROR_RETURN	異常終了です。

例

メモリを新たに確保し、画像を入力し、何らかの処理を実行し、メモリを解放します。

```
#include "f_lsens.h"

void input()
  {
    int mem_no, cmem_no;
    char *adrs;

    cmem_no = Lib_lsensor_get_bin_memory();
    if( ERROR_RETURN != (mem_no = Lib_lsensor_alloc_bin_memory() ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_change_bin_memory( mem_no ) )
          {
            if( NORMAL_RETURN == Lib_lsensor_video_input() )
              {
                Lib_lsensor_video_transmit( 1, BIN_PLANE );
                adrs = Lib_lsensor_adrs_bin_memory( mem_no );
                /*何らかの処理*/
              }
          }
        Lib_lsensor_change_bin_memory( cmem_no );
        Lib_lsensor_free_bin_memory( mem_no );
      }
  }
```

留意事項

○確保されるメモリサイズ（バイト数）は
　　メモリサイズ ＝ 水平取込長 × 垂直取込長 ÷ ８
○ラインセンサカメラ入力の初期化後は、Lib_alloc_bin_memory も同機能となります。

Lib_lsensor_adrs_gray_memory

機　能

濃淡メモリのベース・アドレス通知

形　式

```
#include "f_lsens.h"
char *Lib_lsensor_adrs_gray_memory ( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_alloc_gray_memory で取得した濃淡メモリ用ファイルのベースポインタを取得します。

① ***memory_no*** はメモリＮｏです。
　－１を指定した場合はカレントメモリ用ファイルです。

戻り値

処理結果

値	定　数	意　味
０	NULL	異常終了です。
０以外		濃淡メモリファイルのポインタです。

例

メモリを新たに確保し、画像を入力し、何らかの処理を実行し、メモリを解放します。

```
#include "f_lsens.h"
void input()
  {
    int mem_no, cmem_no;
    char *adrs;

    cmem_no = Lib_lsensor_get_gray_memory();
    if( ERROR_RETURN != (mem_no = Lib_lsensor_alloc_gray_memory() ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_change_gray_memory( mem_no ) )
          {
            if( NORMAL_RETURN == Lib_lsensor_video_input() )
              {
                Lib_lsensor_video_transmit( 1, GRAY_PLANE );
                adrs = Lib_lsensor_adrs_gray_memory( mem_no );
                /*何らかの処理*/
              }
          }
        Lib_lsensor_change_gray_memory( cmem_no );
        Lib_lsensor_free_gray_memory( mem_no );
      }
  }
```

留意事項

○ラインセンサカメラ入力の初期化後は、Lib_adrs_gray_memory も同機能となります。

Lib_lsensor_adrs_bin_memory

機　能

２値メモリのベース・アドレス通知

形　式

```
#include "f_lsens.h"
char *Lib_lsensor_adrs_bin_memory ( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_alloc_bin_memory で取得した２値メモリ用ファイルのベースポインタを取得します。

① ***memory_no*** はメモリＮｏです。
　－１を指定した場合はカレントメモリ用ファイルです。

戻り値

処理結果

値	定　数	意　味
０	NULL	異常終了です。
０以外		２値メモリファイルのポインタです。

例

メモリを新たに確保し、画像を入力し、何らかの処理を実行し、メモリを解放します。

```
#include "f_lsens.h"
void input()
  {
    int mem_no, cmem_no;
    char *adrs;

    cmem_no = Lib_lsensor_get_bin_memory();
    if( ERROR_RETURN != (mem_no = Lib_lsensor_alloc_bin_memory() ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_change_bin_memory( mem_no ) )
          {
            if( NORMAL_RETURN == Lib_lsensor_video_input() )
              {
                Lib_lsensor_video_transmit( 1, BIN_PLANE );
                adrs = Lib_lsensor_adrs_bin_memory( mem_no );
                /*何らかの処理*/
              }
          }
        Lib_lsensor_change_bin_memory( cmem_no );
        Lib_lsensor_free_bin_memory( mem_no );
      }
  }
```

留意事項

○ラインセンサカメラ入力の初期化後は、Lib_adrs_bin_memory も同機能となります。

Lib_lsensor_free_gray_memory

機　能

濃淡メモリの解放

形　式

```
#include "f_lsens.h"
int Lib_lsensor_free_gray_memory( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_alloc_gray_memory で取得した濃淡メモリ用ファイルを、メモリＮｏにより解放します。
カレントメモリ用ファイルは解放できません。

① ***memory_no*** はメモリＮｏです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

メモリを新たに確保し、画像を入力し、何らかの処理を実行し、メモリを解放します。

```
#include "f_lsens.h"
void input()
  {
    int mem_no, cmem_no;
    char *adrs;

    cmem_no = Lib_lsensor_get_gray_memory();
    if( ERROR_RETURN != (mem_no = Lib_lsensor_alloc_gray_memory() ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_change_gray_memory( mem_no ) )
          {
            if( NORMAL_RETURN == Lib_lsensor_video_input() )
              {
                Lib_lsensor_video_transmit( 1, GRAY_PLANE );
                adrs = Lib_lsensor_adrs_gray_memory( mem_no );
                /*何らかの処理*/
              }
          }
        Lib_lsensor_change_gray_memory( cmem_no );
        Lib_lsensor_free_gray_memory( mem_no );
      }
  }
```

留意事項

○ラインセンサカメラ入力の初期化後は、Lib_free_gray_memory も同機能となります。

Lib_lsensor_free_bin_memory

機　能

２値メモリの解放

形　式

```
#include "f_lsens.h"
int  Lib_lsensor_free_bin_memory( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_alloc_bin_memory で取得した２値メモリ用ファイルを、メモリＮｏにより解放します。
カレントメモリ用ファイルは解放できません。

① ***memory_no*** はメモリＮｏです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

メモリを新たに確保し、画像を入力し、何らかの処理を実行し、メモリを解放します。

```
#include "f_lsens.h"
void input()
  {
    int mem_no, cmem_no;
    char *adrs;

    cmem_no = Lib_lsensor_get_bin_memory();
    if( ERROR_RETURN != (mem_no = Lib_lsensor_alloc_bin_memory() ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_change_bin_memory( mem_no ) )
          {
            if( NORMAL_RETURN == Lib_lsensor_video_input() )
              {
                Lib_lsensor_video_transmit( 1, BIN_PLANE );
                adrs = Lib_lsensor_adrs_bin_memory( mem_no );
                /*何らかの処理*/
              }
          }
        Lib_lsensor_change_bin_memory( cmem_no );
        Lib_lsensor_free_bin_memory( mem_no );
      }
  }
```

留意事項

○ラインセンサカメラ入力の初期化後は、Lib_free_bin_memory も同機能となります。

Lib_lsensor_get_gray_memory

機　能

濃淡カレントメモリＮｏの参照

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_gray_memory( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

濃淡メモリのカレントメモリＮｏを参照します。

戻り値

カレントメモリＮｏを返します。

例

カレントメモリＮｏを参照し、そのＮｏが０以外の場合は０に変更します。

```
#include "f_lsens.h"

void input()
  {
    if( 0 != Lib_lsensor_get_gray_memory() )
      {
        Lib_lsensor_change_gray_memory( 0 );
      }
  }
```

留意事項

○ラインセンサカメラ入力の初期化後は、Lib_get_gray_memory も同機能となります。

Lib_lsensor_get_bin_memory

機　能

２値カレントメモリＮｏの参照

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_bin_memory( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

２値メモリのカレントメモリＮｏを参照します。

戻り値

カレントメモリＮｏを返します。

例

カレントメモリＮｏを参照し、そのＮｏが０以外の場合は０に変更します。

```
#include "f_lsens.h"

void input()
  {
    if( 0 != Lib_lsensor_get_bin_memory() )
      {
        Lib_lsensor_change_bin_memory( 0 );
      }
  }
```

留意事項

○ラインセンサカメラ入力の初期化後は、Lib_get_bin_memory も同機能となります。

Lib_lsensor_change_memory

機　能

濃淡＆２値メモリの切り換え

形　式

```
#include "f_lsens.h"
int Lib_lsensor_change_memory ( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

濃淡メモリと２値メモリのカレントメモリを変更します。

① ***memory_no*** はメモリＮｏです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

カレントメモリを変更し、画像を入力します。

```
#include "f_lsens.h"

void input()
  {
    if( NORMAL_RETURN == Lib_lsensor_change_memory( 1 ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_video_input() )
          {
            Lib_lsensor_video_transmit( 1, GRAY_PLANE );
          }
      }
  }
```

留意事項

○画像入力を起動中、この関数を実行した場合の動作は保証しません。
○ラインセンサカメラ入力の初期化後は、Lib_change_memory も同機能となります。

Lib_lsensor_change_gray_memory

機　能

濃淡メモリの切り換え

形　式

```
#include "f_lsens.h"
int Lib_lsensor_change_gray_memory ( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

濃淡メモリのカレントメモリを変更します。

① ***memory_no*** はメモリＮｏです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

カレントメモリを変更し、画像を入力します。

```
#include "f_lsens.h"

void input()
  {
    if( NORMAL_RETURN == Lib_lsensor_change_gray_memory( 1 ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_video_input() )
          {
            Lib_lsensor_video_transmit( 1, GRAY_PLANE );
          }
      }
  }
```

留意事項

○割り込み画像入力を起動中、この関数を実行した場合の動作は保証しません。
○ラインセンサカメラ入力の初期化後は、Lib_change_gray_memory も同機能となります。

Lib_lsensor_change_bin_memory

機　能

２値メモリの切り換え

形　式

```
#include "f_lsens.h"
int Lib_lsensor_change_bin_memory ( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

２値メモリのカレントメモリを変更します。

① ***memory_no*** はメモリＮｏです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

カレントメモリを変更し、画像を入力します。

```
#include "f_lsens.h"

void input()
  {

    if( NORMAL_RETURN == Lib_lsensor_change_bin_memory( 1 ) )
      {
        if( NORMAL_RETURN == Lib_lsensor_video_input() )
          {
            Lib_lsensor_video_transmit( 1, BIN_PLANE );
          }
      }
  }
```

留意事項

○割り込み画像入力を起動中、この関数を実行した場合の動作は保証しません。
○ラインセンサカメラ入力の初期化後は、Lib_change_bin_memory も同機能となります。

Lib_lsensor_bin_memory_cls

機　能

２値メモリクリア

形　式

```
#include "f_lsens.h"
int Lib_lsensor_bin_memory_cls( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

２値メモリの指定メモリＮｏをクリアします。

① ***memory_no*** はメモリＮｏです。
　－１を指定した場合はカレントメモリ用ファイルです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

フレーム・バッファに画像を転送後、メモリをクリアします。

```
#include "f_lsens.h"

void input()
  {
    if( NORMAL_RETURN == Lib_lsensor_video_input() )
      {
        Lib_lsensor_video_transmit( 0, BIN_PLANE );
        Lib_lsensor_bin_memory_cls( 0 );
      }
  }
```

留意事項

○この関数を実行してもフレーム・バッファ上は何ら変化はありません。
　つまり、モニタＴＶの２値画像表示は変化しません。
○ラインセンサカメラ入力の初期化後は、Lib_bin_memory_cls も同機能となります。

Lib_lsensor_gray_memory_cls

機　能

濃淡メモリクリア

形　式

```
#include "f_lsens.h"
int Lib_lsensor_gray_memory_cls( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

濃淡メモリの指定メモリＮｏをクリアします。

① ***memory_no*** はメモリＮｏです。
　－１を指定した場合はカレントメモリ用ファイルです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

フレーム・バッファに画像を転送後、メモリをクリアします。

```
#include "f_lsens.h"

void input()
  {
    if( NORMAL_RETURN == Lib_lsensor_video_input() )
      {
        Lib_lsensor_video_transmit( 0, GRAY_PLANE );
        Lib_lsensor_gray_memory_cls( 0 );
      }
  }
```

留意事項

○この関数を実行してもフレーム・バッファ上は何ら変化はありません。
　つまり、モニタＴＶの濃淡画像表示は変化しません。
○ラインセンサカメラ入力の初期化後は、Lib_gray_memory_cls も同機能となります。

Lib_lsensor_video_transmit

機　能

フレーム・バッファへの画像転送

形　式

```
#include "f_lsens.h"
int Lib_lsensor_video_transmit( int memory_no, int flag );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

指定メモリ上の画像データをフレーム・バッファへ転送します。
５１２×４８０より大きな画像データは縮小して表示します。

① ***memory_no*** はメモリＮｏです。
－１を指定した場合はカレントメモリ用ファイルです。

② ***flag*** は転送メモリフラグです。

値	定　数	意　味
0x01	BIN_PLANE	２値メモリを転送します。
0x20	GRAY_PLANE	濃淡メモリを転送します。

戻り値

処理結果

値	定　数	意　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

画像を入力後、フレーム・バッファに画像を転送します。

```
#include "f_lsens.h"

void input()
  {
    if( NORMAL_RETURN == Lib_lsensor_video_input() )
      {
        Lib_lsensor_video_transmit( 0, GRAY_PLANE );
      }
  }
```

留意事項

○ Lib_ex_video_transmit() の REDUCE_MODE と同じ機能です。
○ Lib_video_transmit() との相違点はラインセンサなどの拡張画像メモリ内容全てをモニタＴＶ上に表示する点です。

Lib_lsensor_get_extract_window

機　能

表示ウインドウ範囲の取得

形　式

```
#include "f_lsens.h"
void Lib_lsensor_get_extract_window( int *xs, int *ys, int *xe, int *ye );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_freeze でフレーム・バッファに転送するときの画像表示ウインドウ位置を取得します。

① ***xs*** はＸ始点位置です。

② ***ys*** はＹ始点位置です。

③ ***xe*** はＸ終点位置です。

④ ***ye*** はＹ終点位置です。

画像表示ウインドウ位置を設定するには後述の、Lib_lsensor_set_extract_window, Lib_lsensor_extraction_display_memory を使用してください。

戻り値

ありません。

例

画像表示ウインドウを原点に変更する。

```
#include "f_lsens.h"

void set()
  {
        int xs, ys, xe, ye;

        Lib_lsensor_get_extract_window( &xs, &ys, &xe, &ye );

        if( ( 0 != xs ) && ( 0 != ys ) )
          {
                xs = 0;
                ys = 0;
                xe = Lib_get_dx_size ( ) - 1;
                ye = Lib_get_dy_size ( ) - 1;
                Lib_lsensor_set_extract_window( xs, ys, xe, ye );
          }
  }
```

留意事項

○ Lib_lsensor_init を実行前にコールする必要があります。
○ Lib_get_extract_window とは同機能です。

Lib_lsensor_set_extract_window

機　能

表示ウインドウの設定

形　式

```
#include "f_lsens.h"
int  Lib_lsensor_set_extract_window( int xs, int ys, int xe, int ye );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_freeze でフレーム・バッファに転送するときの画像表示ウインドウ位置を設定します。

① ***xs*** はＸ始点位置です。

② ***ys*** はＹ始点位置です。

③ ***xe*** はＸ終点位置です。

④ ***ye*** はＹ終点位置です。

画像表示ウインドウ位置を設定するにはこの関数以外に
Lib_lsensor_extraction_display_memory があります。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

画像表示ウインドウを原点に変更する。

```
#include "f_lsens.h"

void set()
  {
        int xs, ys, xe, ye;

        Lib_lsensor_get_extract_window( &xs, &ys, &xe, &ye );

        if( ( 0 != xs ) && ( 0 != ys ) )
          {
                xs = 0;
                ys = 0;
                xe = Lib_get_dx_size ( ) - 1;
                ye = Lib_get_dy_size ( ) - 1;
                Lib_lsensor_set_extract_window( xs, ys, xe, ye );

          }
  }
```

留意事項

○ Lib_lsensor_init を実行前にコールする必要があります。
○ Lib_set_extract_window とは同機能です。

Lib_lsensor_extraction_display_memory

機　能

縮小画像からの画像選択表示

形　式

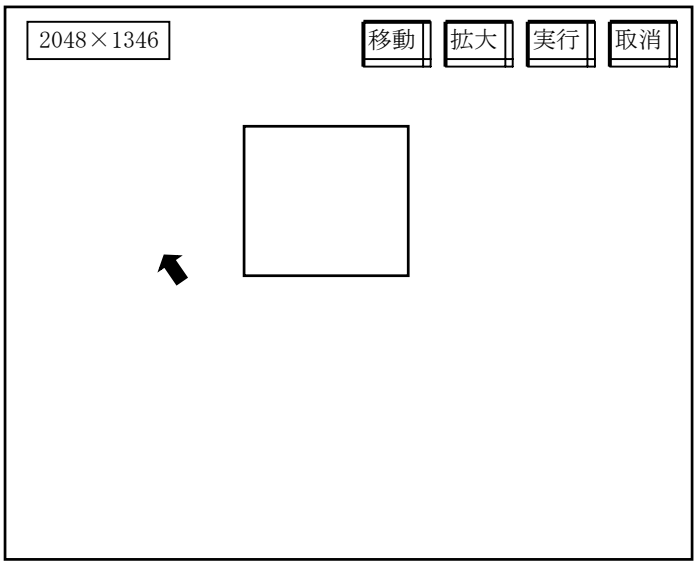
```
#include "f_lsens.h"
int Lib_lsensor_extraction_display_memory( int memory_no );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

指定したメモリ番号の画像をモニタＴＶ上にメモリ全体が表示するように縮小表示し、
モニタＴＶ上で、５１２×４８０の範囲を選択し、通常表示します。
５１２×４８０より大きな画像の場合は縮小して表示します。

① ***memory_no*** はメモリ番号です。（０～）



トラックボール操作

○移動（MOVE）
　画像表示用のウインドウを移動します。
　移動を中止するときは実行キーを押してください。

○拡大（ZOOM）
　画像表示用のウインドウ部分を通常表示に拡大します。
　拡大時もトラックボールにより移動できます。
　中止するときは実行キーを押してください。

○実行（EXEC）
　画像表示ウインドウ位置をシステムパラメータに登録し、終了します。

○取消（CANC）
　終了します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了です。
－１	ERROR_RETURN	異常終了です。

例

初期化後、画像表示ウインドウを実行します。

```
#include "f_lsens.h"

void init()
  {
       Lib_init_cursor();
       if( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
         {
              Lib_lsensor_video_input();
              Lib_lsensor_extraction_display_memory( CURRENT_MEMORY );
         }
  }
```

留意事項

○ Lib_lsensor_init を実行前にコールする必要があります。
○ Lib_init_cursor を実行前にコールする必要があります。
○この関数を実行する前にメモリ上に画像を入力しておく必要があります。
○ Lib_extraction_display_memory と同機能です。
○登録した画像表示ウインドウ位置は Lib_get_extract_window で取得できます。
○ Lib_input_video_control で指定されたプレーン（２値または濃淡のどちらかのみ）を表示します。

Lib_lsensor_adjust_bin_level

機　能

２値レベルの調整

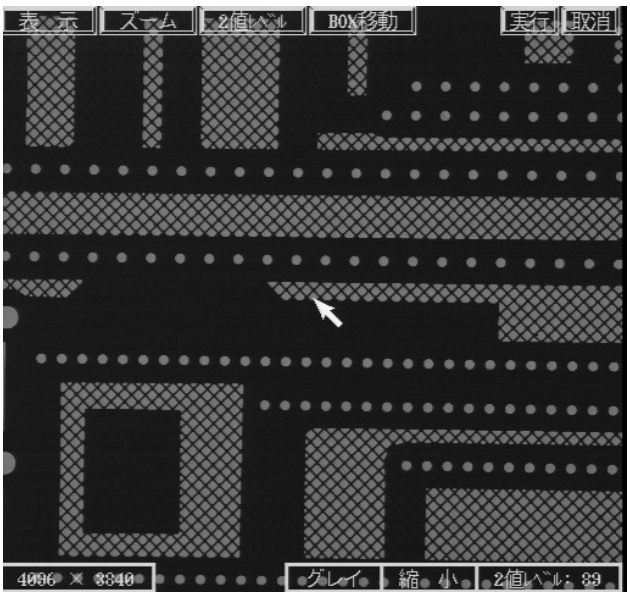
形　式

```
#include "f_lsens.h"
int Lib_lsensor_adjust_bin_level( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

カレントメモリ（グレイ）の画像全体をモニタＴＶに縮小表示します。
トラックボールの操作により、２値レベルの調整を行えます。



トラックボール操作

○表示
　グレイ表示と２値表示を切り替えます。

○ズーム
　拡大表示と縮小表示を切り替えます。

○２値レベル
　２値レベルを調整します。調整を終了するには実行キーを押します。

○ＢＯＸ移動
　縮小表示時に画像表示範囲のウィンドウを移動します。
　拡大表示時もトラックボールにより表示範囲を移動できます。
　移動を終了するには実行キーを押します。

○実行
　２値レベルをシステムパラメータに登録し、終了します。

○取消
　終了します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	カレントメモリがｍ×ｎの拡張メモリではありません。

例

初期化後、２値レベルを調整します。

```
#include "f_lsens.h"

void set_bin_level( void )
{
    Lib_init_cursor();

    if ( NORMAL_RETURN == Lib_lsensor_init( 0, 58, 2108, 0, 2048, 1920, 1 ) )
    {
        Lib_lsensor_video_input();
        Lib_lsensor_adjust_bin_level();
    }
}
```

留意事項

○ Lib_lsensor_init を実行前にコールする必要があります。
○ Lib_init_cursor を実行前にコールする必要があります。
○この関数を実行する前にメモリ上に画像を入力しておく必要があります。

Lib_lsensor_memory_reset

機　能

画像入力メモリのリセット

形　式

```
#include "f_lsens.h"
int Lib_lsensor_memory_reset( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_lsensor_xvideo_input にて変更したラインセンサ画像入力メモリＮＯをカレントＮＯにリセットする。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

Lib_lsensor_xvideo_input の例を参照してください。

留意事項

ありません。

Lib_lsensor_SetGrabMode

機　能

取り込みモードの設定

形　式

```
#include "f_lsens.h"
int  Lib_lsensor_SetGrabMode(int iGrabMode );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

本関数は、ＦＨＣ３３０Ａ，ＦＨＣ３３１０(Ａ)で使用可能です。
画像を取り込むモードを設定します。Lib_lsensor_init 後に呼び出してください。

［ＦＨＣ３３０Ａ］

① ***iGrabMode*** は、画像を取り込むモードです。

値	意　味
０	ノーマルモード
１	トリガモード
２	エンコーダモード

［ＦＨＣ３３１０(Ａ)］

① ***iGrabMode*** は、画像を取り込むモードです。

値	意　味
０	ノーマルモード
１	トリガモード
２	エンコーダモード
３	エンコーダライン選択モード

ノーマル、トリガ、エンコーダの各モードの違いは、留意事項を参照してください。

戻り値

処理結果

値	定　数	意　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

```
#include "f_lsens.h"
int  EncoderLineSelectModeForFHC3310( void )
{
    const intENCODER_LINESELECT_MODE = 3;
    return Lib_lsensor_SetGrabMode( ENCODER_LINESELECT_MODE );

}
```

留意事項　○Lib_lsensor_init 後は、ノーマルモードに設定されます。

○画像入力モード

［ノーマルモード］

ライントリガは、設定されたスキャンレートで常時出力され続けます。
画像取り込みを起動した時に、次のライントリガから、指定されたライン数の画像を入力します。

［トリガモード］

ライントリガは、設定されたスキャンレートで常時出力され続けます。
画像取り込みを起動した後に、外部トリガを待ち、外部トリガの立ち下がりエッジを検出すると、次のライントリガから、指定されたライン数の画像を入力します。

［エンコーダモード］

ロータリエンコーダの信号によりエンコーダカウンタが動作し、比較レジスタとの一致によりライントリガが出力されます。
Ｚ相、外部トリガによるエンコーダカウンタのリセット機能を使用する場合、カウンタの動作開始は、Ｚ相、外部トリガを待ちます。

各モードの詳細な使い方は「ＦＨＣ３３０Ａ取扱説明書」「ＦＨＣ３３１０(Ａ)取扱説明書」をご覧ください。

○902_HR2.LIB(IPM-8500A 用)には、本関数が存在しません。リンク時にリンクエラーになります。

（ラインセンサカメラ制御ライブラリサンプルコード１）

ＦＨＣ３３１０(A)ラインセンサ連続取込＆中断。画面表示は取り込みと非同期で行ないます。

```
#define false  0
#define true   !false
#define BUFFER_NUM    4

int ContinuousGrab(void)
{
    const int       INFINITY = 0;
    char            string[256];
    int             ret,loop,i,x,y,memories[BUFFER_NUM];
    unsigned int    lin=0,img=0,mem=0;
    double          tim;

    //キャプチャーバージョン確認
    if( 3 > Lib_lsensor_hardver() )      return ERROR_RETURN;
    //連続取込用グレイメモリ確保
    for( ret = NORMAL_RETURN , i = 0 ; i < BUFFER_NUM ; i++ )
        if( ERROR_RETURN == ( memories[i] = Lib_alloc_gray_memory() ) )
            ret = ERROR_RETURN;
    if( NORMAL_RETURN == ret )
    {
        //連続取込開始（無限）
        ret = Lib_lsensor_continuous_grab(INFINITY,BUFFER_NUM,memories);
        if( NORMAL_RETURN == ret )
        {
            Lib_strtclk_count();
            for( loop = true ; loop ; )
            {
                if( CURSOR_CANCEL == Lib_see_current_position(&x,&y) )
                {
                    Lib_lsensor_video_input_abort();//取込中断処理
                    loop = false;
                }
                img = Lib_lsensor_get_grab_count();//取込枚数取得
                lin = Lib_lsensor_get_line_count();//取込ライン数取得
                mem = img % BUFFER_NUM; //取込先メモリインデックス
                //縮小表示
                Lib_ex_video_transmit(mem,GRAY_PLANE,REDUCE_MODE);
                tim = Lib_readclk_count() * 0.000813802;
                if( loop )    Lib_chrdisp(20,29,"<< right click to abort >>");
                else          Lib_chrdisp(20,29,"                          ");
                sprintf(string,"Image:%06d Line:%6d Memory:%1d  time:%9.3fs"
                        ,img,lin,memories[mem],tim);
                Lib_chrdisp( 1,30,string);
            }
        }
        //連続取込用グレイメモリ破棄
        for( i = 0 ; i < BUFFER_NUM ; i++ )  Lib_free_gray_memory( memories[i] );
    }
    return ret;
}
```

Lib_lsensor_SetGrabCondition

機　能

諸設定

形　式

```
#include "f_lsens.h"
int  Lib_lsensor_SetGrabCondition( iPhase, iP1, iP2, iP3, iP4, iP5 );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

本関数は、ＦＨＣ３３０Ａ，ＦＨＣ３３１０(Ａ)で使用可能です。
画像取り込みのための設定を行います。Lib_lsensor_init 後に呼び出してください。

① ***iPhase*** はクロックに対するビデオの位相反転を行います。
カメラからの画像にノイズが発生する場合は、この設定で調整してください。

値	意　　味
0	同位相
1	反転

② ***iP1*** は将来用です。現状０を必ず指定してください。

③ ***iP2*** は将来用です。現状０を必ず指定してください。

④ ***iP3*** は将来用です。現状０を必ず指定してください。

⑤ ***iP4*** は将来用です。現状０を必ず指定してください。

⑥ ***iP5*** は将来用です。現状０を必ず指定してください。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

```
#include "f_lsens.h"

void SetGrabMode( void )
{
    if ( NORMAL_RETURN == Lib_lsensor_init( 4, 58, 2108, 0, 2048, 1920, 0 ) )
    {
        Lib_lsensor_SetGrabCondition( 1, 0, 0,
                                      0, 0, 0 );    /* 反転に設定します。 */

        /* 処理 */
    }
}
```

留意事項

○Lib_lsensor_init 後は、同位相に設定されます。
○902_HR2.LIB(IPM-8500A 用)には、本関数が存在しません。リンク時にリンクエラーになります。

Lib_lsensor_SetEncoder

機　能

エンコーダ入力の設定

形　式

```
#include "f_lsens.h"
int  Lib_lsensor_SetEncoder( iSignal, iSampling, iZ, iExtTrigger,
                             iReverse, uiCmp1, uiCmp2, iDelayLine );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

本関数は、ＦＨＣ３３０ＡとＦＨＣ３３１０(A)で使用可能です。
エンコーダを使用し、画像取り込みをするための設定を行います。Lib_lsensor_init 後に呼び出してください。

［ＦＨＣ３３０Ａ］

① ***iSignal*** はエンコーダの入力規格です。
ＲＳ－４２２，オープンコレクタ（フォトカプラ）入力を選択します。

値	意　味
0	ＲＳ－４２２
1	オープンコレクタ（フォトカプラ）

② ***iSampling*** はエンコーダ信号のサンプリング速度です。

値	意　味
0	通常
1	２倍
2	４倍

③ ***iZ*** はエンコーダＺ相入力の有効／無効設定です。
Ｚ相を有効にするとエンコーダカウンタのカウント開始、エンコーダカウンタのリセットが有効になります。

値	意　味
0	無効
1	有効

④ ***iExtTrigger*** は外部トリガの有効／無効設定です。
外部トリガを有効にするとエンコーダカウンタのカウント開始、エンコーダカウンタのリセットが有効になります。

値	意　味
0	無効
1	有効

⑤ ***iReverse*** はパルスカウンタのＵＰ，ＤＯＷＮの反転指定です。

値	意　味
0	通常
1	反転

⑥ ***uiCmp1*** は比較レジスタ１です。
ここで設定した値とエンコーダカウンタが一致した場合、ライントリガを出力します。
０以外を設定してください。
⑦ ***uiCmp2*** は比較レジスタ２です。
ここで設定した値とエンコーダカウンタが一致した場合、ライントリガを出力します。
設定範囲は１～６５５３５です。

⑧ ***iDelayLine*** は、遅延ライン数の設定です。
最初のライントリガが出力されるまでのＣＣＤセンサの光電荷の過飽和状態による不定画像を、取り込まないようにするための遅延ライン数を設定します。
設定範囲は０～１０２３です。

［ＦＨＣ３３１０(A)］
① ***iSignal*** はエンコーダの入力規格です。
ＦＨＣ３３１０(A)はＲＳ－４２２のみ対応しています。０を指定して下さい。

値	意　味
０	ＲＳ－４２２

② ***iSampling*** はエンコーダ信号のサンプリング速度です。

値	意　味
１	２倍
２	４倍

③ ***iZ*** はエンコーダＺ相入力の有効／無効設定です。
Ｚ相を有効にするとエンコーダカウンタのカウント開始、エンコーダカウンタのリセットが有効になります。

値	意　味
０	無効
１	有効

④ ***iExtTrigger*** は外部トリガの有効／無効設定です。
ＦＨＣ３３１０は対応していません。０を指定して下さい。
ＦＨＣ３３１０Ａは対応しています。

値	意　味
０	無効
１	有効

⑤ ***iReverse*** はパルスカウンタのＵＰ，ＤＯＷＮの反転指定です。

値	意　味
０	通常
１	反転

⑥ ***uiCmp1*** は比較レジスタ１です。
ここで設定した値とエンコーダカウンタが一致した場合、ライントリガを出力します。
設定範囲は０～４２９４９６７２９５です。
⑦ ***uiCmp2*** は比較レジスタ２です。
ここで設定した値とエンコーダカウンタが一致した場合、ライントリガを出力します。
設定範囲は１～６５５３５です。

⑧ ***iDelayLine*** は、遅延ライン数の設定です。
最初のライントリガが出力されるまでのＣＣＤセンサの光電荷の過飽和状態による不定画像を、取り込まないようにするための遅延ライン数を設定します。
設定範囲は０～１０２３です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

```
#include "f_lsens.h"
void EncoderSetting( void )
{
    if ( NORMAL_RETURN == Lib_lsensor_init( 4, 58, 2108, 0, 2048, 1920, 0 ) )
    {
        if ( NORMAL_RETURN == Lib_lsensor_SetEncoder( 0, 2, 0, 0, 0, 0, 4, 1 ) )
        {
                /* 処理 */
         }
    }
}
```

留意事項

○本関数の設定項目の詳細は「ＦＨＣ３３０Ａ取扱説明書」「ＦＨＣ３３１０(A)取扱説明書」をご覧ください。

○902_HR2.LIB(IPM-8500A 用)には、本関数が存在しません。リンク時にリンクエラーになります。

○FHC3310(A)をお使いの場合は、三菱レイヨン(株)、竹中システム機器(株)のラインセンサカメラのみエンコーダ入力可能となります。

Lib_lsensor_hardver

機　能

キャプチャカードハードウェアバージョン取得

形　式

```
#include "f_lsens.h"
short Lib_lsensor_hardver ( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

キャプチャカードのハードウェアバージョンを取得します。
ラインセンサ初期化後に呼び出してください。

戻り値

処理結果

値	定　　数	意　　味
０～		ハードウェアバージョン
－１	ERROR_RETURN	異常終了しました。

例

ラインセンサカメラ制御ライブラリサンプルコード１を参照して下さい。

留意事項

○本関数は、ＦＨＣ３３１０(A)のみで使用可能です。
○ＦＨＣ３３１０(A)は下記のバージョンがあります。(2002/06/26)

値	機　能
０	単発取込
３	取込中断・連続取込

Lib_lsensor_video_input_abort

機　能　取込中断・停止処理

形　式

```
#include "f_lsens.h"
void Lib_lsensor_video_input_abort(void);
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説　カメラからの映像取込を中断し、取込動作を停止します。

戻り値　ありません。

例　ラインセンサカメラ制御ライブラリサンプルコード１を参照して下さい。

留意事項　○本関数は、ＦＨＣ３３１０（バージョン３以降）及びＦＨＣ３３１０Ａで使用可能です。

Lib_lsensor_continuous_grab

機　能

ラインセンサ連続取込開始

形　式

```
#include "f_lsens.h"
int Lib_lsensor_continuous_grab( int image_num, int memory_num,
                                 int* memory_index);
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

取込枚数・取込先メモリ（複数指定可）を指定して、連続取込を開始します。
Lib_lsensor_init後に呼び出してください。

① ***image_num*** は取込枚数を指定します。

値	意　　味
０	無限枚数連続取込
１	単発取込（Lib_lsensor_xvideo_inputと同等です）
２～１６３８３	指定枚数取込

② ***memory_num*** は取込先メモリ枚数です。③の配列数を指定して下さい。

③ ***memory_index*** は取込先メモリ番号配列の先頭アドレスです。
memory_num個のint型配列に取込先グレイメモリ番号を設定して下さい。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

ラインセンサカメラ制御ライブラリサンプルコード１を参照して下さい。

留意事項

○本関数は、ＦＨＣ３３１０（バージョン３以降）及びＦＨＣ３３１０(A)で使用可能です。

○image_num == 0 もしくは image_num > memory_num の場合は
取込先メモリは循環します。（リングバッファ動作）

○指定枚数連続取込の終了は、Lib_lsensor_get_video_status で確認して下さい。
終了確認後は必ず、Lib_lsensor_video_input_abort で停止処理をして下さい。
停止処理を行わないと、他の関数が異常終了します。

○無限枚数連続取込は、Lib_lsensor_video_input_abort で停止して下さい。

Lib_lsensor_get_grab_count

機　能　ラインセンサ取込確定枚数取得

形　式

```
#include "f_lsens.h"
unsigned int Lib_lsensor_get_grab_count( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説　取込中に、取込確定枚数を取得します。
ラインセンサカメラ取込開始後に呼び出してください。

戻り値　取込確定枚数（０～）

例　ラインセンサカメラ制御ライブラリサンプルコード１を参照して下さい。

留意事項　○本関数は、ＦＨＣ３３１０（バージョン３以降）及びＦＨＣ３３１０(Ａ)で使用可能です。

Lib_lsensor_get_line_count

機　能　ラインセンサ取込ライン数取得

形　式

```
#include "f_lsens.h"
unsigned int Lib_lsensor_get_line_count( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説　取込中に、取込ライン数を取得します。
ラインセンサカメラ取込開始後に呼び出してください。

戻り値　取込確定枚数（０～）

例　ラインセンサカメラ制御ライブラリサンプルコード１を参照して下さい。

留意事項　○本関数は、ＦＨＣ３３１０（バージョン３以降）及びＦＨＣ３３１０(A)で使用可能です。

Lib_lsensor_create_param

機　能

ラインセンサ用キャプチャパラメータの生成

形　式

```
#include "f_lsens.h"
void *Lib_lsensor_create_param( int param_id );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

キャプチャパラメータＩＤで指定されたキャプチャパラメータ用構造体を生成します。

① ***param_id*** は生成するキャプチャパラメータのＩＤ番号です。

値	定　　数	意　　味	対応キャプチャ
0x10002	CAPTPRM_LSENS_FHC3310_ID	ラインセンサ用定義	FHC3310(A)

＊キャプチャパラメータＩＤに対応するキャプチャパラメータ用構造体に関しては
後述の留意事項をごらんください。

戻り値

処理結果

値	意　　味
0	異常終了しました
0 以外	キャプチャパラメータへのポインタです

例

ラインセンサカメラ制御ライブラリサンプルコード２，３を参照してください。

留意事項　○キャプチャパラメータＩＤに対応する構造体定義を以下に示します。

【ラインセンサ用定義１】

```
typedef struct
{
  unsigned int      uiPhase;          /*  サンプリングクロックの位相      */
  unsigned int      uiCameraClock;    /*  カメラ駆動クロック              */
  unsigned int      uiSamplingClock;  /*  サンプリングクロック            */
  unsigned int      uiDelayX;         /*  Ｘ方向無効画素数                */
  unsigned int      uiDelayY;         /*  Ｙ方向無効画素数                */
  unsigned int      uiImageX;         /*  Ｘ方向画像サイズ                */
  unsigned int      uiImageY;         /*  Ｙ方向画像サイズ                */
  unsigned int      uiScanRate;       /*  蓄積時間                        */
  unsigned int      uiExt_Trg;        /*  外部トリガ                      */
  int               iStoreMode;       /*  メモリ格納形式                  */
  unsigned int      iCamBpp;          /*  カメラ出力ビット数              */
} CAPTPRM_LSENS_FHC3310_T;
```

① uiPhase はサンプリングクロックの位相設定です。
カメラからの画像にノイズが発生する場合は、この設定で調整してください。

値	意　　味
0	同位相
1	位相反転

② uiCameraClock はカメラ駆動クロックです。
使用するカメラに合わせて設定してください。
設定可能な数値についてはサンプリングクロックの説明を参照してください。

③ uiSamplingClock はサンプリングクロックです。
カメラ駆動クロックと同じ設定値にすると、クロック対ビデオ比が１：１となります。

値	意　　味
40000000	40MHZ
20000000	20MHZ
10000000	10MHZ
5000000	5MHZ

④ uiDelayX はＸ方向無効画素数です。
設定範囲は 0～16383 です。

⑤ uiDelayY はＹ方向無効画素数です。
設定範囲は 0～1023 です。

⑥ uiImageX はＸ方向画像サイズです。
カメラの有効画素数を指定してください。
設定範囲は 32～16383 です。また、ここで指定する値は 32 の倍数である必要があります。

⑦ uiImageY はＹ方向画像サイズです。
設定範囲は 1～65535 です。

⑧ uiScanRate は蓄積時間(スキャンレート)です。
最小蓄積時間(１スキャンあたりの最小必要時間)以上を画素数で指定します。
設定範囲は(uiDelayX + uiImageX)～65535 です。

⑨ uiExt_Trg は外部トリガ設定です。

定　　数	意　　味
OFF	外部トリガを使用しません
ON	外部トリガを使用します

⑩ iStoreMode はメモリ格納形式です。

定　　数	意　　味
BITxx_MODE	カメラ出力ビット数に従う
BIT8_MODE	1画素 8Bit
BIT10_MODE	1画素 10Bit　(注)902 では使用できません

⑪ iCamBpp はカメラ出力ビット数です。
使用するカメラに合わせて設定してください。

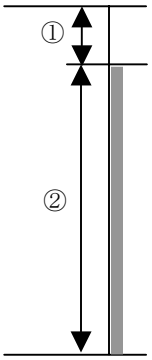
値	意　　味
8	1画素 8Bit
10	1画素 10Bit

各パラメータと取り込み範囲の関係は以下のようになります。

（X方向画像取り込み範囲）

①X方向無効画素数
②X方向画像サイズ
③蓄積時間（スキャンレート）
④（X方向無効画素数）←自動計算されます

（Y方向画像取り込み範囲）



①Y方向無効画素数
②Y方向画像サイズ

Lib_lsensor_destroy_param

機　能

キャプチャパラメータの破棄

形　式

```
#include "f_lsens.h"
int Lib_lsensor_destroy_param( void *param );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

指定されたキャプチャパラメータ用のメモリを解放します。

① ***param*** は破棄するキャプチャパラメータへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

ラインセンサカメラ制御ライブラリサンプルコード２，３を参照してください。

留意事項

ありません

Lib_lsensor_set_param

機　能

キャプチャパラメータの設定

形　式

```
#include "f_lsens.h"
int Lib_lsensor_set_param( void *param );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

カレントチャネルのキャプチャパラメータを設定します。

① ***param*** は設定するキャプチャパラメータへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

ラインセンサカメラ制御ライブラリサンプルコード２，３を参照してください。

留意事項

○実装されているキャプチャと、キャプチャパラメータの対応キャプチャが一致しない場合はエラーとなります。

Lib_lsensor_get_param

機　能

キャプチャパラメータの取得

形　式

```
#include "f_lsens.h"
int Lib_lsensor_get_param( void *param );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

カレントチャネルのキャプチャパラメータを取得します。

① ***param*** は取得するキャプチャパラメータへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

ラインセンサカメラ制御ライブラリサンプルコード２，３を参照してください。

留意事項

○実装されているキャプチャと、キャプチャパラメータの対応キャプチャが一致しない場合はエラーとなります。

（ラインセンサカメラ制御ライブラリサンプルコード２）

ＴＬ－７４５０ＵＦＤ（竹中システム機器製）を使用する場合の例を示します。

```
int lsens_init_for_tl7450ufd( void )
{
  CAPTPRM_LSENS_FHC3310_T     *tpLsens1;
  int          re;

  re = ERROR_RETURN;
  if(NULL!=(tpLsens1= (CAPTPRM_LSENS_FHC3310_T *)Lib_lsensor_create_param
                                                    ( CAPTPRM_LSENS_FHC3310_ID )))
  {
    tpLsens1->uiPhase          = 0;          /*  サンプリングクロックの位相 : 同位相       */
    tpLsens1->iCameraClock     = 40000000;   /*  カメラ駆動クロック         : 40MHZ        */
    tpLsens1->iSamplingClock   = 40000000;   /*  サンプリングクロック       : 40MHZ        */
    tpLsens1->uiScanRate       = 10000;      /*  蓄積時間[clock]            : 10000        */
    tpLsens1->uiDelayX         = 210;        /*  Ｘ方向無効画素             : 210          */
    tpLsens1->uiDelayY         = 0;          /*  Ｙ方向無効画素             : 0            */
    tpLsens1->uiImageX         = 7424;       /*  Ｘ方向画像サイズ           : 7424         */
    tpLsens1->uiImageY         = 1024;       /*  Ｙ方向画像サイズ           : 1024         */
    tpLsens1->uiExt_Trg        = OFF;        /*  外部トリガ                 : 使用しない   */
    tpLsens1->iStoreMode       = BIT8_MODE;  /*  メモリ格納形式             : １画素 8Bit  */
    tpLsens1->iCamBpp          = 10;         /*  カメラ出力ビット数         : １画素 10Bit */
    re = Lib_lsensor_set_param( tpLsens1 );

    Lib_lsensor_destroy_param( tpLsens1 );
  }

  return( re );
}
```

（ラインセンサカメラ制御ライブラリサンプルコード３）

現在の設定を取得し、Ｙ方向画像サイズのみを 480 に変更する場合の例を示します。

```
int lsens_resize_y( void )
{
  CAPTPRM_LSENS_FHC3310_T       *tpLsens1;
  int           re;

  re = ERROR_RETURN;
  if(NULL!=(tpLsens1= (CAPTPRM_LSENS_FHC3310_T *)Lib_lsensor_create_param
                                                    ( CAPTPRM_LSENS_FHC3310_ID )))
  {
    if( NORMAL_RETURN == Lib_lsensor_get_param( tpLsens1 ) )
    {
      tpLsens1->uiImageY = 480;        /*  Ｙ方向画像サイズ          : 480   */
      re = Lib_lsensor_set_param( tpLsens1 );
    }
    Lib_lsensor_destroy_param( tpLsens1 );
  }

  return( re );
}
```

6. 高分解能エリアセンサ制御ライブラリ

本ライブラリは高分解能エリアセンサを使用するためのものです。
各ライブラリの注意事項をよく読み、正しくお使いください。

２００２年１０月現在のサポート情報
弊社で標準サポートしている高分解能エリアセンサは

［ＦＨＣ３３１］［ＦＨＣ３３２１］
・ＦＣ１３００　（竹中システム機器　製）
・ＦＣ１３２０Ｖ　（竹中システム機器　製）
・ＫＰ－Ｆ１００　（日立国際電気　製）
・ＣＳ３９１０　（東京電子工業　製）
・ＣＳ３９２０　（東京電子工業　製）

［ＦＨＣ３３１０(A)］
・ＫＰ－Ｆ１２０　（日立国際電気　製）
・ＳＫＣ－１３３　（池上通信機　製）
・ＦＣ１５００Ｆ　（竹中システム機器　製）

の８種類です。

6.1　本ライブラリをご使用の前に

９０２，ＦＶＬ／ＬＮＸにて本ライブラリを使用するには、以下のハードウェアが必要です。

キャプチャボード

当社でサポートしている高分解能エリアカメラ用キャプチャボードは以下の３種類です。

◆（株）グラフイン
　ハイレゾエリアカメラＩ／Ｆボード　　**ＩＰＭ－８５００Ｄ**

◆（株）ファースト
　デジタルカメラキャプチャボード　　**ＦＨＣ３３１，ＦＨＣ３３１ＬＶ**
　　ＦＨＣ３３２１

◆（株）ファースト
　高速・高分解能　画像入力ボード　　**ＦＨＣ３３１０(Ａ)**

なお上記装置の取り付けについては、キャプチャボード提供メーカから
提供される説明書を良くご覧の上、**慎重に取り付けてください。**

※注意　ＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１を外部トリガモードでお使いの場合、画像取り込み関数発行時にトリガ信号がＬｏｗの状態であると、トリガが入力されたと判定します。

高分解能エリアカメラ

当社でサポートしている高分解能エリアカメラは以下の８種類です。

◆竹中システム機器（株）提供

フルフレームシャッタビデオカメラ	ＦＣ１３００
デジタルケーブル	後述参照
ＲＳ２３２Ｃケーブル	ＲＳ２３－０２
カメラ電源	ＰＵ－９７
カメラケーブル	１２Ｗ－０２

フルフレームシャッタビデオカメラ	ＦＣ１３２０Ｖ
デジタルケーブル	後述参照
カメラ電源	ＰＵ－９７
カメラケーブル	１２Ｗ－０２

注：ＦＣ１３２０ＶはＦＨＣ３３１ＬＶまたはＦＨＣ３３２１との組み合わせでのみ使用可能です。

フルフレームシャッタビデオカメラ	ＦＣ１５００Ｆ
デジタルケーブル	後述参照
カメラ電源	ＰＵ１００
カメラケーブル	１２Ｗ－０２

注：ＦＣ１５００ＦはＦＨＣ３３１０との組み合わせでのみ使用可能です。

◆日立国際電気（株）提供

全画素読みだし白黒ＣＣＤカメラ	ＫＰ－Ｆ１００
デジタルケーブル	後述参照
ＡＣアダプタ	ＡＰ－６０Ａ（改）
カメラケーブル	Ｃ－２０１ＫＳＭ
ジャンクションボックス	ＪＵ－Ｍ１Ａ

全画素読みだし近赤外高解像度ＣＣＤカメラ	ＫＰ－Ｆ１２０
デジタルケーブル	後述参照
ＡＣアダプタ	ＡＰ－６０Ａ（改）
カメラケーブル	Ｃ－２０１ＫＳＭ
ジャンクションボックス	ＪＵ－Ｍ１Ａ

注：ＫＰ－Ｆ１２０はＦＨＣ３３１０との組み合わせでのみ使用可能です。

◆東京電子工業（株）提供

高解像度ＣＣＤカメラ	ＣＳ３９１０
デジタルケーブル	後述参照
ＡＣアダプタ	ＣＡ１３０Ｂ
電源ケーブル	ＣＰＣ３９１０－０３

高解像度ＣＣＤカメラ	ＣＳ３９２０
デジタルケーブル	後述参照
ＡＣアダプタ	ＣＡ１３０Ｂ
電源ケーブル	ＣＰＣ３９１０－０３

注：ＣＳ３９１０、ＣＳ３９２０はＦＨＣ３３１ＬＶまたはＦＨＣ３３２１との組み合わせでのみ使用可能です。

◆池上通信機（株）提供

高解像度白黒フルフレームシャッタカメラ	ＳＫＣ－１３３
デジタルケーブル	後述参照
ＡＣアダプタ	ＡＣＰ－１３０
ＤＣケーブル	ＰＳＣ１３０－３Ａ

注：ＳＫＣ－１３３はＦＨＣ３３１０との組み合わせでのみ使用可能です。

カメラ用デジタルケーブル

カメラ用デジタルケーブルは下記のカメラ用デジタルケーブル対応表をご覧の上で、必要な製品をご用意ください。

カメラ用デジタルケーブル対応表

	ＩＰＭ－８５００Ｄ	ＦＨＣ３３１	ＦＨＣ３３１ＬＶ ＦＨＣ３３２１
FC1300	竹中システム機器（株） **DG36-02-8500D8(2m)** **DG36-05-8500D8(5m)**	竹中システム機器（株） **DG36-02AV(2m)** **DG36-05AV(5m)**	竹中システム機器（株） **DG36-02AV(2m)** **DG36-05AV(5m)**
FC1320V	使用不可	使用不可	竹中システム機器（株） **DG36-02AV(2m)** **DG36-05AV(5m)**
KP-F100	(株) グラフイン **HITA-26300(3m)**	(株)ファースト **F100CBL-330AH(2m)**	株)ファースト **F100CBL-330AH(2m)**
CS3910	使用不可	使用不可	東京電子工業(株) **CDC3910D2-03(3m)** **CDC3910D2-05(5m)**
CS3920	使用不可	使用不可	東京電子工業(株) **CDC3910D2-03(3m)** **CDC3910D2-05(5m)**

	ＦＨＣ３３１０(Ａ)
KP-F120	(株) アバールデータ **CBL-Z028(3m)**
SKC-133	池上通信機（株） **RSC130-3A2(3m)** **RSC130-5A2(5m)**
FHC1500F	竹中システム機器（株） **DG36-02APC(2m)** **DG36-05APC(5m)**

6.2　カメラ側の設定

本項では、高分解能エリアセンサを使用し画像を取り込むためのカメラ設定を記載しています。

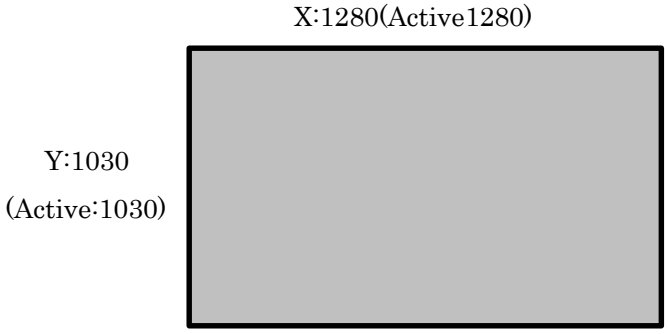
ＦＣ１３００、ＦＣ１３２０Ｖ

注意：ＦＣ１３２０ＶはＦＨＣ３３１ＬＶ、ＦＨＣ３３２１との組み合わせで使用可能です。

◆取り込み画像サイズ
画像取り込みサイズは横１２８０×縦１０３０画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１２７９，１０２９）



（注：倍速読み出し時は（０，０）－（１２７９，５１４）の範囲が感光されている領域となります。）

◆サポートしている機能
・連続電子シャッター動作、ランダムシャッター動作
・通常読み出し、倍速読み出し　（注１、注２）
・Exposure(露光時間)は任意に設定できます。　（注３）
・ゲインは任意に設定できます。

注１：倍速読み出しは、連続シャッター使用時のみ有効です。
注２：ＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１使用時は通常読み出しのみ対応しております。
注３：高速、低速シャッターのどちらもサポートしています。

◆重要な設定事項

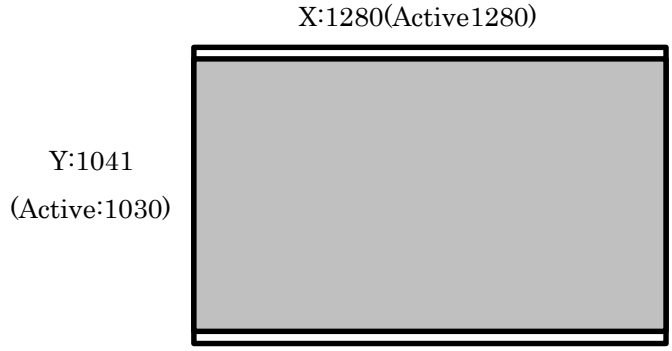
ＩＰＭ－８５００Ｄを使用する場合は、**ＦＤＶ発生位相の変更**を行ってください。ＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１を使用する場合はデフォルトのままで結構です。

設定方法およびカメラ仕様につきましては竹中システム機器株式会社より提供される「ＦＣ１３００（ＦＣ１３２０）　オペレーションマニュアル」を参照してください。

［ＩＰＭ－８５００Ｄ使用時］

◆取り込み画像サイズ
画像取り込みサイズは横１２８０×縦１０４１画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの内、感光されている領域（カメラから取り込まれた有効な画像領域）は座標位置（０，２）－（１２７９，１０３１）の範囲です。



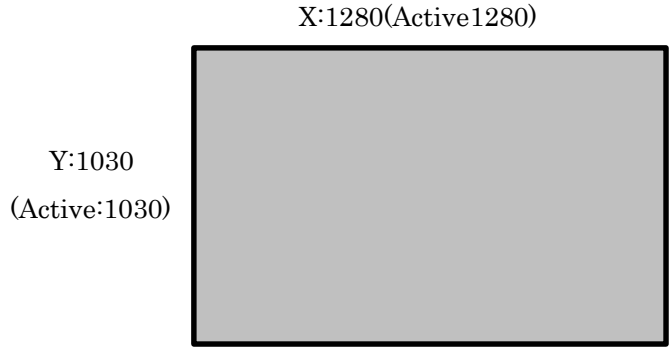
注：読み出し速度１／２４のモードでは（０，０）－（１２７９，５１４）の範囲が感光されている領域となります。

［ＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１使用時］

◆取り込み画像サイズ
画像取り込みサイズは横１２８０×縦１０３０画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１２７９，１０２９）の範囲です。



◆サポートしている機能
・ＦＤ（フレーム・オン・デマンド）機能　注１
・シャッタースピードは任意に設定できます。
・シャッターモードはＦＤ機能のＯＮ、ＯＦＦによって以下のように設定できます。
（ＦＤがＯＦＦのとき）
１：電子シャッターＯＦＦ　２：電子シャッターＯＮ
（ＦＤがＯＮのとき）
１：固定シャッター
・ＲＤスイッチ（読み出し速度切替え）は任意に設定可能です。　注２
（注：電子シャッターＯＮ時は読み出し速度１／１２で使用してください）

注１：ＦＤ（フレーム・オン・デマンド）はＩＰＭ－８５００Ｄでは使用できません。
注２：読み出し速度切替えはＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１では使用できません。

設定方法およびカメラ仕様につきましては日立国際電気株式会社より提供されるＫＰ－Ｆ１００取扱説明書を参照してください。

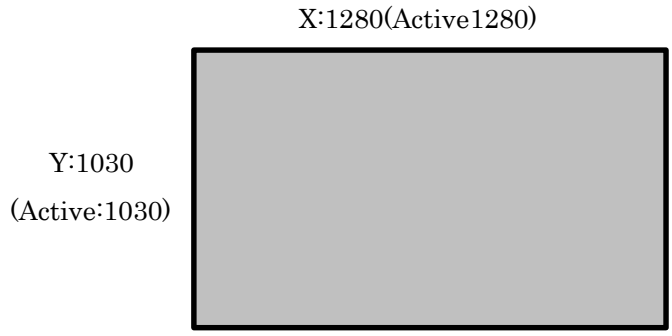
ＣＳ３９１０

注意：本カメラはＦＨＣ３３１ＬＶ、ＦＨＣ３３２１との組み合わせでのみ使用可能です。

◆取り込み画像サイズ
画像取り込みサイズは横１２８０×縦１０３０画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１２７９，１０２９）



◆サポートしている機能
・ノーマル電子シャッタ動作、ランダムトリガシャッタ動作
・１２Ｈｚ全画素読み出し　（注１）
・シャッタスピードは任意に設定できます。

注１：２４Ｈｚ倍速読み出しはサポートしておりません。

設定方法およびカメラ仕様につきましては東京電子工業株式会社より提供される「ＣＳ３９１０　取扱説明書」を参照してください。

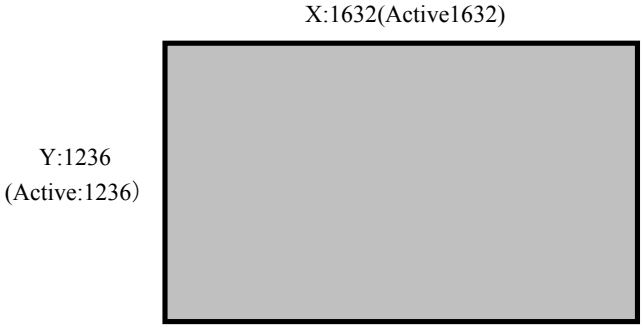
ＣＳ３９２０

注意：本カメラはＦＨＣ３３１ＬＶ、ＦＨＣ３３２１との組み合わせでのみ使用可能です。

◆取り込み画像サイズ
画像取り込みサイズは横１６３２×縦１２３６画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１６３１，１２３５）



◆サポートしている機能
・ランダムトリガシャッタ動作
・７．５Ｈｚフレーム読み出し　（注１）
・１５Ｈｚフィールド読み出し　（注２）
・シャッタスピードは任意に設定できます。

注１：フレーム読み出し時は、露光動作を２回行うため動画像の取得には適していません。
注２：フィールド読み出し時は、縦方向画像サイズが半分になります。

設定方法およびカメラ仕様につきましては東京電子工業株式会社より提供される「ＣＳ３９２０　機能仕様書」を参照してください。

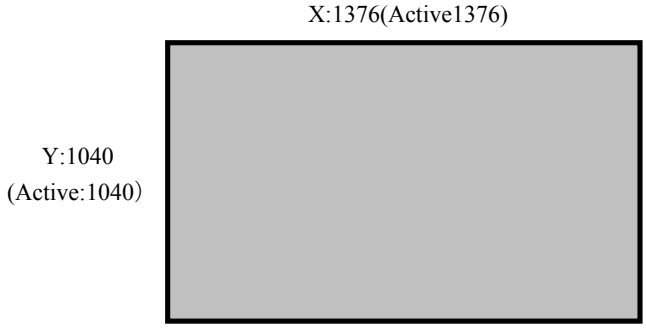
ＫＰ－Ｆ１２０

注意：本カメラはＦＨＣ３３１０(Ａ)との組み合わせでのみ使用可能です。

◆取り込み画像サイズ
画像取り込みサイズは横１３７６×縦１０４０画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１３７５，１０３９）



◆サポートしている機能
・ランダムトリガシャッタ動作
・シャッタスピードは任意に設定できます。

設定方法およびカメラ仕様につきましては株式会社日立国際電気より提供される「ＫＰ－Ｆ１２０　機能仕様書」を参照してください。

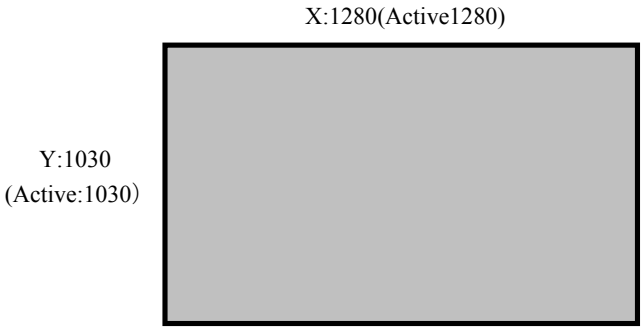
ＳＫＣ－１３３

注意：本カメラはＦＨＣ３３１０(A)との組み合わせでのみ使用可能です。

◆取り込み画像サイズ
画像取り込みサイズは横１２８０×縦１０３０画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１２７９，１０２９）



◆サポートしている機能
・ランダムトリガシャッタ動作
・シャッタスピードは任意に設定できます。

設定方法およびカメラ仕様につきましては池上通信機株式会社より提供される「ＳＫＣ－１３３　オペレーションマニュアル」を参照してください。

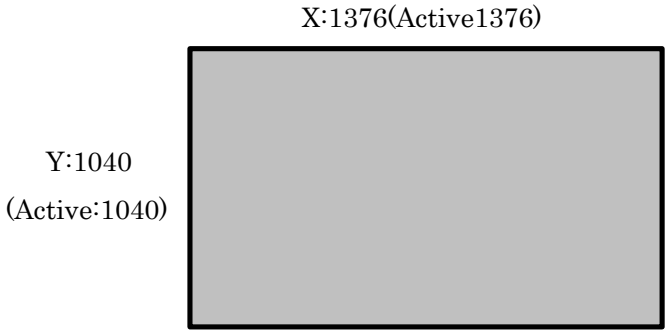
ＦＣ１５００Ｆ

注意：本カメラはＦＨＣ３３１０との組み合わせでのみ使用可能です。

◆取り込み画像サイズ
画像取り込みサイズは横１３７６×縦１０４０画素です。

◆取り込み画像感光画素エリア
取り込み画像サイズの全てが感光されている領域（カメラから取り込まれた有効な画像領域）です。

座標位置（０，０）－（１３７５，１０３９）



◆サポートしている機能
・連続電子シャッター動作、ランダムシャッター動作
　・Exposure(露光時間)は任意に設定できます。
　・ゲインは任意に設定できます。

設定方法およびカメラ仕様につきましては竹中システム機器株式会社より提供される「ＦＣ１５００Ｆ　取扱説明書」を参照してください。

Lib_high_res_init

機　能

高分解能エリアセンサ画像取り込み処理の初期化

形　式

```
#include "f_hres.h"
int Lib_high_res_init( int mode );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

映像入力をＲＳ１７０ビデオ入力から高分解能エリアセンサ入力に切り換え、高分解能画像取り込みのための初期化を行います。

① ***mode***（ カメラ種別 ）

（９０２シリーズの場合）

値	定　　数	意　　味
０	FC1300	FC1300,FC1320Vをトリガ入力で使用します。
２	FC1300EXT	FC1300,FC1320Vを外部トリガ入力で使用します。
６	FC1300NRM	FC1300,FC1320Vを通常入力で使用します。
３	KP_F100	KP-F100をトリガ入力で使用します。
５	KP_F100EXT	KP-F100を外部トリガ入力で使用します。
７	KP_F100NRM	KP-F100を通常入力で使用します。
８	CS3910	CS3910をトリガ入力で使用します。
９	CS3910EXT	CS3910を外部トリガ入力で使用します。
１０	CS3910NRM	CS3910を通常入力で使用します。
１７	CS3920	CS3920（フレーム読み出し）をトリガ入力で使用します。
１８	CS3920EXT	CS3920（フレーム読み出し）を外部トリガ入力で使用します。
１９	CS3920_FLD	CS3920（フィールド読み出し）をトリガ入力で使用します。
２０	CS3920EXT_FLD	CS3920（フィールド読み出し）を外部トリガ入力で使用します。
３２	SKC_133	SKC-133をトリガ入力で使用します。
３３	SKC_133EXT	SKC-133を外部トリガ入力で使用します。
３４	SKC_133NRM	SKC-133を通常入力で使用します。
３５	KP_F120	KP-F120をトリガ入力で使用します。
３６	KP_F120EXT	KP-F120を外部トリガ入力で使用します。
３７	KP_F120NRM	KP-F120を通常入力で使用します。
３８	FC1500F	FC1500Fをトリガ入力で使用します。
３９	FC1500FEXT	FC1500Fを外部トリガ入力で使用します。
４０	FC1500FNRM	FC1500Fを通常入力で使用します。

注１：キャプチャボードにより使用不可能なパラメータがあります。
　　詳細は留意事項の（カメラ種別対応表）を参照してください。
注２：トリガ入力、外部トリガ入力に関しては、留意事項を参照してください。

（ＦＶＬ／ＬＮＸの場合）

値	定　　数	意　　味
１００	FC1300	FC1300,FC1320Vをトリガ入力で使用します。
１０１	FC1300EXT	FC1300,FC1320Vを外部トリガ入力で使用します。
１０２	FC1300NRM	FC1300,FC1320Vを通常入力で使用します。
１０３	KP_F100	KP-F100をトリガ入力で使用します。
１０４	KP_F100EXT	KP-F100を外部トリガ入力で使用します。
１０５	KP_F100NRM	KP-F100を通常入力で使用します。
１０６	CS3910	CS3910をトリガ入力で使用します。
１０７	CS3910EXT	CS3910を外部トリガ入力で使用します。
１０８	CS3910NRM	CS3910を通常入力で使用します。
１１５	CS3920	CS3920（フレーム読み出し）をトリガ入力で使用します。
１１６	CS3920EXT	CS3920（フレーム読み出し）を外部トリガ入力で使用します。
１１８	CS3920_FLD	CS3920（フィールド読み出し）をトリガ入力で使用します。
１１９	CS3920EXT_FLD	CS3920（フィールド読み出し）を外部トリガ入力で使用します。
２００	SKC_133	SKC-133をトリガ入力で使用します。
２０１	SKC_133EXT	SKC-133を外部トリガ入力で使用します。
２０２	SKC_133NRM	SKC-133を通常入力で使用します。
２０３	KP_F120	KP-F120をトリガ入力で使用します。
２０４	KP_F120EXT	KP-F120を外部トリガ入力で使用します。
２０５	KP_F120NRM	KP-F120を通常入力で使用します。
２０６	FC1500F	FC1500Fをトリガ入力で使用します。
２０７	FC1500FEXT	FC1500Fを外部トリガ入力で使用します。
２０８	FC1500FNRM	FC1500Fを通常入力で使用します。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。
－１	ARG_ERR	カメラ種別が不明です。
－１０	UNUSUAL_ERR	ボードまたは９０２の初期化に失敗しました。
－９１	NO_DEVICE	ボードが存在していません。

例

（高分解能エリアセンサ制御ライブラリサンプルコード１）を参照してください。

留意事項

○画像取り込を開始する前に、本関数をコールしてください。

○キャプチャボードを２枚使用する場合は本関数をコールする前に、チャネル番号の指定を行ってください。

○カメラ種別でトリガ入力を指定した場合は、取り込み関数（Lib_high_res_freeze, Lib_high_res_xfreeze）をコールした時点でカメラに映像取り込み指示を出します。
通常入力との相違点は、映像取り込みのタイミングを指定できる点です。そのため撮像物が移動している場合でも、確実にカメラの視野内に撮像物をとらえることができます。
また、トリガ入力をする場合はカメラ側をトリガーモードにする必要があるのでご注意ください。
カメラの設定方法・キャプチャボードとの接続などは、ＦＨＣ３３１０(A)取扱説明書，ＦＨＣ３３１取扱説明書，ＦＨＣ３３２１取扱説明書またはＩＰＭ－８５００取扱手引書を参照してください。

○外部トリガ入力は、上述の映像取り込みのタイミングを画像処理装置以外の装置にまかせる場合に使用します。
トリガ入力との相違点は、取り込み関数をコールしても画像取り込みの準備をするだけという点です。その後、外部の装置からのトリガを受けてはじめて画像取り込みを開始します。
また、外部トリガ入力をする場合もカメラ側をトリガモードにする必要があります。
カメラの設定方法、キャプチャボードとの接続などはＦＨＣ３３１０(A)取扱説明書，ＦＨＣ３３１取扱説明書またはＩＰＭ－８５００取扱手引書を参照してください。

○ＦＨＣ３３１ＬＶ，ＦＨＣ３３２１をお使いの場合、トリガ入力、外部トリガ入力時の取り込み時間は、約　露光時間＋８５ｍｓです。
通常入力時の取り込み時間は、約８５ｍｓ～１６８ｍｓです。

○ＣＳ３９２０をお使いの場合
　フィールド読み出し時：露光時間＋６７ｍｓ
　フレーム読み出し時　：露光時間＋１３５ｍｓ
となります。

○ＦＨＣ３３１０(A)をお使いの場合
　通常入力　　：約３４ｍｓ～６８ｍｓ
　トリガ入力　　：約　露光時間＋３４ｍｓ
　外部トリガ入力　　：約　露光時間＋３４ｍｓ
となります。

○キャプチャと使用可能なカメラ種別の対応は以下のようになります。
（カメラ種別対応表）

カメラ種別	IPM-8500D	FHC331	FHC331LV FHC3321	FHC3310(A)
FC1300	○	△	○	×
FC1300EXT	×	△	○	×
FC1300NRM	○	△	○	×
KP_F100	×	○	○	×
KP_F100EXT	×	○	○	×
KP_F100NRM	○	○	○	×
CS3910	×	×	○	×
CS3910EXT	×	×	○	×
CS3910NRM	×	×	○	×
CS3920	×	×	○	×
CS3920EXT	×	×	○	×
CS3920_FLD	×	×	○	×
CS3920EXT_FLD	×	×	○	×
SKC_133	×	×	×	○
SKC_133EXT	×	×	×	○
SKC_133NRM	×	×	×	○
KP_F120	×	×	×	○
KP_F120EXT	×	×	×	○
KP_F120NRM	×	×	×	○
FC1500F	×	×	×	○
FC1500FEXT	×	×	×	○
FC1500FNRM	×	×	×	○

（○：使用可能　△：FC1320V は使用不可　×：使用不可）

Lib_high_res_xinit

機　能

高分解能画像取り込み処理の初期化Ⅱ

形　式

```
#include "f_hres.h"
int Lib_high_res_xinit( unsigned int uiImageX, unsigned int uiImageY,
                        unsigned int uiDelayX, unsigned int uiDelayY, int iTrigger,
                                               unsigned int uiExtTrigger, int iMode );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

弊社標準サポート外のカメラを使用する場合に使用します。
標準サポート外のカメラについては、弊社にて動作確認することが出来ませんので、お客様の責任で本関数はご使用ください。
本関数はＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３１０(Ａ)で使用可能です。

［ＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１］

① ***uiImageX*** はＸ方向の有効画素数です。
設定範囲は、５１２～４０６４です。指定する値は３２の倍数である必要があります。

② ***uiImageY*** はＹ方向の有効画素数です。
設定範囲は、１～４０９５です。

③ ***uiDelayX*** はＸディレイです。
ＨＤの立ち上がりから有効画像データまでのクロック数を指定します。
ここでの設定値＋３クロック後の画像から入力を開始します。
設定範囲は、０～２５５です。

④ ***uiDelayY*** はＹディレイです。
ＶＤの立ち上がりから有効画像データまでのクロック数を指定します。
設定範囲は、０～２５５です。

⑤ ***iTrigger*** は画像入力トリガパルス幅です。
画像入力トリガを使用しない場合（通常入力）は、0 を指定してください。
画像入力トリガを使用する場合は、トリガパルス幅をＨＤパルス幅単位で指定します。
ここで設定した値がプラスの場合は正論理で、マイナスの場合は負論理で画像入力トリガパルスが出力されます。

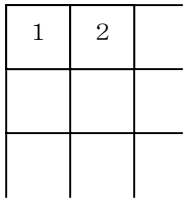
⑥ ***uiExtTrigger*** は外部トリガ入力の使用選択です。
外部トリガ入力を使用する場合は 1 を、使用しない場合は 0 を指定してください。

⑦ ***iMode*** はカメラの出力モードです。

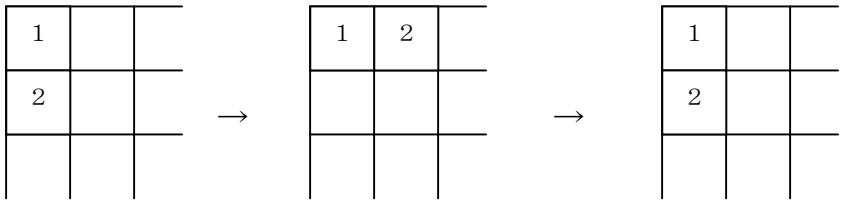
値	定　　数	意　　味
0	BIT8_MODE	１画素８ビット出力カメラ
2	BIT16_MODE_A	横方向２画素１６ビット出力カメラ
3	BIT16_MODE_B	縦方向２画素１６ビット出力カメラ

○１画素８ビット出力カメラ
代表的なカメラとしてＦＣ１３００，ＦＣ１３２０，ＫＰ－Ｆ１００，ＣＳ３９１０があります。

○横方向２画素１６ビット出力カメラ
代表的なカメラとして日立国際電気株式会社ＫＰ－Ｆ１１０があります。
下記のように横１、２を同時に出力するカメラです。



○縦方向２画素出力カメラ
代表的なカメラとしてＫＯＤＡＫ　ＥＳ４．０があります。
下記のように縦１、２を同時に出力するカメラです。



この仕様のカメラを使用する場合は、画像入力後に、Lib_high_res_convert_for_bit16b を使用して画像変換をする必要があります。

［ＦＨＣ３３１０(A)］

① ***uiImageX*** はＸ方向の有効画素数です。
設定範囲は、５１２～１６３６８です。指定する値は３２の倍数である必要があります。

② ***uiImageY*** はＹ方向の有効画素数です。
設定範囲は、１～６５５３５です。

③ ***uiDelayX*** はＸディレイです。
ＨＤの立ち上がりから有効画像データまでのクロック数を指定します。
ＦＨＣ３３１０(A)のハードウェアバージョンによって設定範囲が変わります。

バージョン	設定範囲
０	０～２５５
３	０～１６３８３

ハードウェアバージョンは、Lib_high_res_hardver 関数で確認して下さい。

④ ***uiDelayY*** はＹディレイです。
ＶＤの立ち上がりから有効画像データまでのクロック数を指定します。
設定範囲は、０～６５５３５です。

⑤ ***iTrigger*** は画像入力トリガパルス幅です。
画像入力トリガを使用しない場合（通常入力）は、0 を指定してください。
画像入力トリガを使用する場合は、トリガパルス幅をＨＤパルス幅単位で指定します。
ここで設定した値がプラスの場合は正論理で、マイナスの場合は負論理で画像入力トリガパルスが出力されます。

⑥ ***uiExtTrigger*** は外部トリガ入力の使用選択です。
外部トリガ入力を使用する場合は 1 を、使用しない場合は 0 を指定してください。

⑦ ***iMode*** はカメラの出力モードです。

値	定　数	意　味
０	BIT8_MODE	１画素８ビット出力カメラ
２	BIT16_MODE_A	横方向２画素１６ビット出力カメラ

戻り値

処理結果

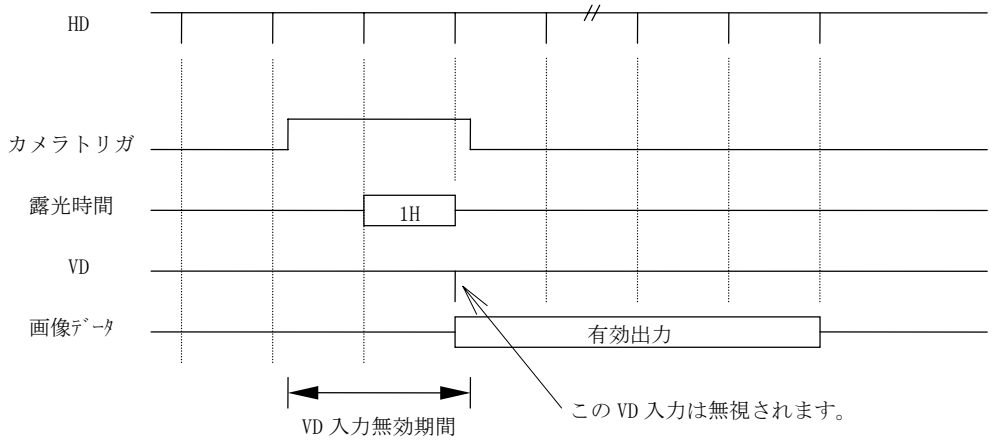
値	定　数	意　味
０	NORMAL_RETURN	正常終了
－１	ERROR_RETURN	異常終了
－１０	UNUSUAL_ERR	ボードまたは９０２の初期化に失敗しました。
－９１	NO_DEVICE	ボードが存在していません。

例

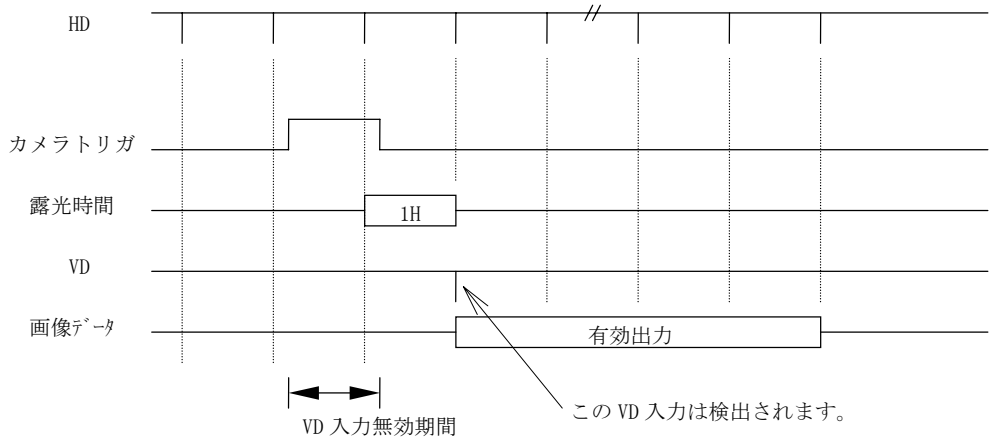
（高分解能エリアセンサ制御ライブラリサンプルコード４）を参照してください。

留意事項　○竹中システム機器株式会社のＦＣ１５００Ｆはこの関数では初期化出来ません。
後述の Lib_high_res_set_param 関数で初期化して下さい。

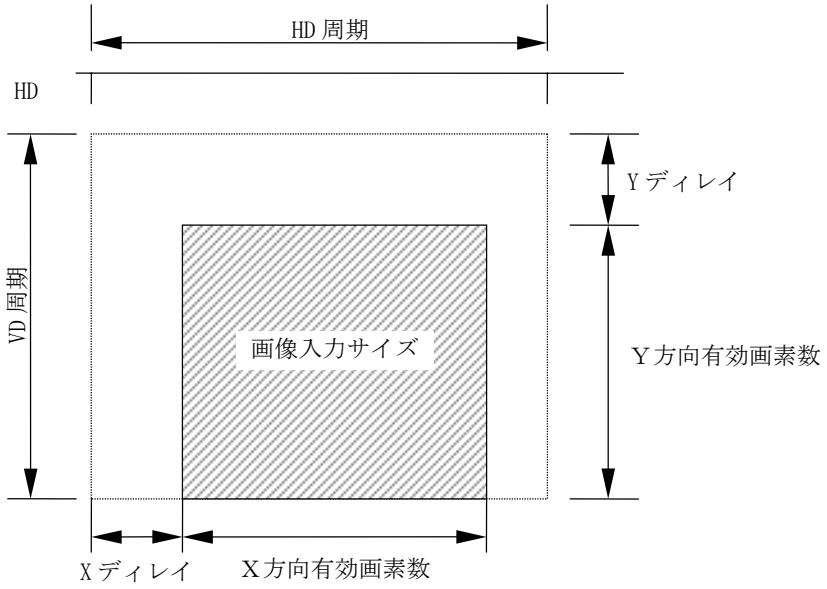
○トリガーモード使用時の注意点
本モジュールがトリガを発生している期間中は、VD の入力を無視します。トリガ終了後に VD の入力が有効になります。トリガを露光スタートとして使用する場合、出力トリガ幅によっては画像入力ができないことがあります。
例えばカメラ側でトリガ後の露光時間を 1”HD”期間と設定した場合は、露光後すぐに VD が出力されますので、トリガ幅を 2”HD”以上に設定すると VD の検出が行われません。



このような場合は、出力トリガ幅を 1”HD”期間としてください。



○X，Yディレイと有効画素数の関係



Lib_high_res_finish

機　能

高分解能エリアセンサ画像取り込み処理の終了

形　式

```
#include "f_hres.h"
int Lib_high_res_finish( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

映像入力を高分解能エリアセンサ入力からＲＳ１７０ビデオ入力に切り換えます。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。
－９９	NOT_INIT	初期化されていません。

例

（高分解能エリアセンサ制御ライブラリサンプルコード１）を参照してください。

留意事項

○ＤＭＡ動作中（画像取り込み中等）に実行した場合は異常終了します。

○ Lib_high_res_init を実行後、ユーザソフトを終了する場合は、必ずこの関数を実行してください。

Lib_high_res_freeze

機　能

高分解能エリアセンサ画像の入力（フリーズ）

形　式

```
#include "f_hres.h"
int Lib_high_res_freeze( int video_control,  int transmit_flag );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

最新の高分解能エリアセンサ入力画像をカレントメモリ上に入力し、転送フラグの指定があれば Lib_set_extract_window で設定している画像表示ウインドウ位置をモニタＴＶ上に表示します。

① ***video_control*** は転送メモリフラグです。

値	定　　数	意　　味
0x01	BIN_PLANE	２値メモリに転送します。
0x20	GRAY_PLANE	濃淡メモリに転送します。

② ***transmit_flag*** は転送フラグです。

値	定　　数	意　　味
1	TRANSMIT	フレーム・バッファに画像を転送します。
0	NOT_TRANSMIT	カレントメモリにのみ画像を転送します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
1	TRANSMITING	画像転送中です。(取り込不可)
－１	ERROR_RETURN	異常終了しました。
－９９	NOT_INIT	初期化されていません。

例

（高分解能エリアセンサ制御ライブラリサンプルコード１）を参照してください。

留意事項

○メモリへの画像入力完了まで終了しません。
○転送フラグに TRANSMIT を指定した場合、Lib_set_extract_window で設定した画像表示ウインドウ部分をフレーム・バッファに転送し、モニタＴＶに表示しています。
○デジタルケーブルを接続していない場合、キャプチャボードによって動作が異なります。（下記参照）
・ＩＰＭ８５００Ｄ使用時は戻り値が NORMAL_RETURN となります。
・ＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１使用時は関数内部から処理が戻ってきません。
・ＦＨＣ３３１０(Ａ)使用時は関数内部から処理が戻ってきません。

Lib_high_res_xfreeze

機　能

高分解能エリアセンサ画像の入力開始

形　式

```
#include "f_hres.h"
int Lib_high_res_xfreeze( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

最新の高分解能エリアセンサ入力画像をカレントメモリ上に入力します。
本ライブラリは Lib_high_res_freeze と違い、起動後、入力完了を待たずに終了します。
取り込みの完了は、後述の Lib_high_res_get_video_status で取得してください。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
１	TRANSMITING	画像転送中です。(取り込不可)
－１	ERROR_RETURN	異常終了しました。
－９９	NOT_INIT	初期化されていません。

例

（高分解能エリアセンサ制御ライブラリサンプルコード２）を参照してください。

留意事項

○ビデオ入力画像は濃淡画像のみです。
　２値画像を取得したい場合は“Lib_binary_convert”等により濃淡画像から
　２値画像への変換転送を行ってください。
○画像入力を起動中、高分解能画像関係の関数を実行した場合の動作は保証しません。

Lib_high_res_get_video_status

機　能

高分解能エリアセンサ画像取り込み完了ステータスの取得

形　式

```
#include "f_hres.h"
int Lib_high_res_get_video_status( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

Lib_high_res_xfreeze 実行後、画像取り込み完了ステータスを取得します。

戻り値

処理結果

値	定　　数	意　　味
０	FINISHED_TRANSMIT	画像入力が完了しました。
１	TRANSMITING	画像入力中です。
－１	ERROR_RETURN	異常終了しました。
－９９	NOT_INIT	初期化されていません。

例

（高分解能エリアセンサ制御ライブラリサンプルコード２）を参照してください。

留意事項

○画像入力を起動後、この関数でステータスを取得しないまま、画像メモリをアクセスした場合、メモリ内容の保証はできません。

Lib_high_res_stop

機　能　高分解能エリアセンサ画像取り込み強制終了

形　式

```
#include "f_hres.h"
void Lib_high_res_stop( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			

解　説　画像取り込み中に取り込みを強制終了させます。

戻り値　ありません。

例　（高分解能エリアセンサ制御ライブラリサンプルコード２）を参照してください。

留意事項　○本関数はＩＰＭ－８５００Ｄをご使用の場合のみ使用できます。

Lib_high_res_set_video_channel

機　能

ビデオ入力チャネル設定

形　式

```
#include "f_hres.h"
void Lib_high_res_set_video_channel( int channel );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

キャプチャボードを２枚実装されている場合には、本関数をコールすることでカレントチャネル（ビデオ入力対象ボード）を切り替えることができます。

① ***channel*** はチャネル番号(０～１)です。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

（高分解能エリアセンサ制御ライブラリサンプルコード２）を参照してください。

留意事項

○本関数はキャプチャボードを２枚実装している場合のみ使用できます。
○本関数を呼び出さない場合のカレントチャネル番号は０です。
○以下の関数は、カレントチャネルに対して動作します。
　Lib_high_res_init, Lib_high_res_finish, Lib_high_res_freeze,
　Lib_high_res_xfreeze, Lib_high_res_get_video_status, Lib_high_res_stop
○画像入力中は、本関数でチャネルの変更は行わないでください。
○チャネル間で画像メモリのサイズが異なる場合は、必ず、高分解能エリアセンサ処理の終了（Lib_high_res_finish）後にチャネルを切り替えてください。
○ＩＰＭ－８５００ＡとＩＰＭ－８５００Ｄは共存が可能ですが、ＩＰＭ－８５００Ａのチャネルに切り替える場合は、必ず、高分解能エリアセンサ処理の終了（Lib_high_res_finish）後にチャネルを切り替えてください。

Lib_high_res_get_video_channel

機　能

ビデオ入力チャネル取得

形　式

```
#include "f_hres.h"
int Lib_high_res_get_video_channel( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

現在のカレントチャネル（ビデオ入力対象ボード）番号を取得します。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

キャプチャボードの枚数を取得し関数値として返します。

```
#include “f_hres.h”

int get_channel( void )
{
        return( Lib_high_res_get_video_channel() );
}
```

留意事項

ありません。

Lib_high_res_SetGrabCondition

機　能

諸設定

形　式

```
#include "f_hres.h"
int  Lib_high_res_SetGrabCondition( iPhase, iP1, iP2, iP3, iP4, iP5 );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			

解　説

本関数は、ＦＨＣ３３２１，ＦＨＣ３３１０(A)で使用可能です。
画像取り込みのための設定を行います。Lib_high_res_init 後に呼び出してください。

① ***iPhase*** はクロックに対するビデオの位相反転を行います。
カメラからの画像にノイズが発生する場合は、この設定で調整してください。

値	意　味
0	同位相
1	反転

② ***iP1*** は将来用です。現状０を必ず指定してください。

③ ***iP2*** は将来用です。現状０を必ず指定してください。

④ ***iP3*** は将来用です。現状０を必ず指定してください。

⑤ ***iP4*** は将来用です。現状０を必ず指定してください。

⑥ ***iP5*** は将来用です。現状０を必ず指定してください。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

```
#include "f_hres.h"

void SetGrabMode( void )
{
    if ( NORMAL_RETURN == Lib_high_res_init( FHC1300 )
    {
        Lib_high_res_SetGrabCondition( 1, 0, 0,
                                       0, 0, 0 );    /* 反転に設定します。 */

        /* 処理 */
    }
}
```

留意事項

○Lib_high_res_init 後は、同位相に設定されます。
○902_HR2.LIB(IPM-8500A 用)には、本関数が存在しません。リンク時にリンクエラーになります。

Lib_high_res_get_video_channel_num

機　能

ビデオ入力チャネル数の取得

形　式

```
#include "f_hres.h"
int Lib_high_res_get_video_channel_num( void );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

ビデオ入力チャネル数（キャプチャボードの枚数）を取得します。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

ビデオ入力チャネル数を取得し関数値として返します。

```
#include “f_hres.h”

int get_channel_num( void )
{
        return( Lib_high_res_get_video_channel_num() );
}
```

留意事項

○ＩＰＭ－８５００ＡとＩＰＭ－８５００Ｄが搭載されている場合のチャネル数は２となります。

Lib_high_res_convert_for_bit16b

機　能

縦方向２画素１６ビット出力カメラ画像の変換

形　式

```
#include "f_hres.h"
int Lib_high_res_convert_for_bit16b( int wDstMem );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

縦方向２画素１６ビット出力カメラ（KODAK ES4.0等）から取り込んだカレントメモリの画像を変換します。
詳細は、Lib_high_res_xinit を参照してください。

① ***wDstMem*** は変換後の画像の出力先グレイメモリＮｏです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

縦方向２画素１６ビット出力カメラから画像を取り込み、画像変換後モニターに表示します。

```
#include "f_stdlib.h"
#include "f_video.h"
#include "f_gui.h"
#include "f_hres.h"

void main( void )
{
        int             wDstMem;



        Lib_init_cursor();
        if( NORMAL_RETURN == Lib_high_res_xinit( 1280, 1030, 50,
                                        0, 0, OFF, BIT16_MODE_B ) )
        {
                if( 0 < ( wDstMem = Lib_alloc_gray_memory() ) )
                {
                        Lib_high_res_freeze( GRAY_PLANE, NOT_TRANSMIT );
                        Lib_high_res_convert_for_bit16b( wDstMem );
                        Lib_ex_video_transmit( wDstMem,
                                        GRAY_PLANE, REDUCE_MODE );

                        Lib_display_keyinput( 0, 450, "確認" );
                        Lib_free_gray_memory( wDstMem );
                }
                Lib_high_res_finish();
        }

}
```

留意事項

○本関数はＦＨＣ３３１、ＦＨＣ３３１ＬＶ、ＦＨＣ３３２１でのみ使用可能です。
○画像出力先メモリにカレントメモリを指定することはできません。

Lib_high_res_create_param

機　能

高精細エリアカメラ用キャプチャパラメータの生成

形　式

```
#include "f_hres.h"
void *Lib_high_res_create_param( int param_id );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

キャプチャパラメータＩＤで指定されたキャプチャパラメータ用構造体を生成します。

① ***param_id*** は生成するキャプチャパラメータのＩＤ番号です。

値	定　数	意　味	対応キャプチャ
0x10001	CAPTPRM_AREACAM_FHC3310_ID	エリアカメラ用定義	FHC3310(A)

＊キャプチャパラメータＩＤに対応するキャプチャパラメータ用構造体に関しては
後述の留意事項をごらんください。

戻り値

処理結果

値	意　味
0	異常終了しました
0 以外	キャプチャパラメータへのポインタです

例

（高分解能エリアセンサ制御ライブラリサンプルコード５）を参照してください。

留意事項　○それぞれのキャプチャパラメータＩＤに対応する構造体定義を以下に示します。

【エリアカメラ用定義１】

```
typedef struct
{
  int               iCamCode;      /*  カメラコード                    */
  unsigned int      uiPhase;       /*  サンプリングクロックの位相      */
  unsigned int      uiDelayX;      /*  Ｘ方向無効画素数                */
  unsigned int      uiDelayY;      /*  Ｙ方向無効画素数                */
  unsigned int      uiImageX;      /*  Ｘ方向画像サイズ                */
  unsigned int      uiImageY;      /*  Ｙ方向画像サイズ                */
  unsigned int      uiTotalX;      /*  Ｘ方向全画素数                  */
  unsigned int      uiTotalY;      /*  Ｙ方向全画素数                  */
  int               iTrigger;      /*  トリガパルス幅                  */
  unsigned int      uiExt_Trg;     /*  外部トリガ                      */
  int               iStoreMode;    /*  メモリ格納形式                  */
  unsigned int      iCamBpp;       /*  カメラ出力ビット数              */
  int               iOutMode;      /*  データ出力方式                  */
} CAPTPRM_AREACAM_FHC3310_T;
```

① iCamCode はカメラコードです。
本パラメータでカメラコードを指定した場合､構造体のその他のメンバにはカメラコード毎のデフォルト値がセットされます。
デフォルト値を変更したい場合は、各メンバの説明を参照して設定してください。

定　　数	意　　味
SKC_133	SKC-133 をトリガ入力で使用します。
SKC_133EXT	SKC-133 を外部トリガ入力で使用します。
SKC_133NRM	SKC-133 を通常入力で使用します。
KP_F120	KP-F120 をトリガ入力で使用します。
KP_F120EXT	KP-F120 を外部トリガ入力で使用します。
KP_F120NRM	KP-F120 を通常入力で使用します。
FC1500F	FC1500F をトリガ入力で使用します。
FC1500FEXT	FC1500F を外部トリガ入力で使用します。
FC1500FNRM	FC1500F を通常入力で使用します。

② uiPhase はサンプリングクロックの位相です。
カメラからの画像にノイズが発生する場合は、この設定で調整してください。

値	意　　味
0	同位相
1	位相反転

③ uiDelayX はＸ方向無効画素数です。
設定範囲は 0～16383 です。

④ uiDelayY はＹ方向無効画素数です。
設定範囲は 0～1023 です。

⑤ uiImageX はＸ方向画像サイズです。
カメラのＸ方向有効画素数を指定してください。
設定範囲は 32～16383 です。また、ここで指定する値は 32 の倍数である必要があります。

⑥ uiImageY はＹ方向画像サイズです。
設定範囲は 1～65520 です。

⑦ uiTotalX はＸ方向全画素数です。
カメラのＸ方向全画素数を指定してください。
設定範囲は(uiDelayX + uiImageX)～16383 です。

⑧ uiTotalY はＹ方向全画素数です。
カメラのＸ方向全画素数を指定してください。
設定範囲は(uiDelayY + uiImageY)～65535 です。

⑨ iTrigger はトリガパルス幅です。
トリガを使用しない場合は 0 に設定してください。
また、正の値を設定すると正論理、負の値を設定すると負論理となります。

⑩ uiExt_Trg は外部トリガです。

定　　数	意　　味
OFF	外部トリガを使用しません
ON	外部トリガを使用します

⑪ iStoreMode はメモリ格納形式です。

定　　数	意　　味
BITxx_MODE	カメラ出力ビット数に従う
BIT8_MODE	１画素 8Bit
BIT10_MODE	１画素 10Bit
BIT16_MODE_A	１Ｈ毎に１ライン出力(上位８ビットは隣の画素)

⑫ iCamBpp はカメラ出力ビット数です。
使用するカメラに合わせて設定してください。

値	意　　味
8	1 画素 8Bit
10	1 画素 10Bit

⑬ iOutMode はデータ出力方式です。
使用するカメラに合わせて設定してください。

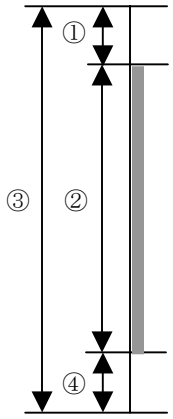
定　　数	意　　味
PROG_MODE	プログレッシブ
INTER_MODE	インターレース

各パラメータと取り込み範囲の関係は以下のようになります。

（X方向画像取り込み範囲）

①X方向無効画素数
②X方向画像サイズ
③X方向全画素数
④(X方向無効画素数)←自動計算されます

（Y方向画像取り込み範囲）



①Y方向無効画素数
②Y方向画像サイズ
③Y方向全画素数
④(Y方向無効画素数)←自動計算されます

Lib_high_res_destroy_param

機　能

キャプチャパラメータの破棄

形　式

```
#include "f_hres.h"
int Lib_high_res_destroy_param( void *param );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

指定されたキャプチャパラメータ用のメモリを解放します。

① ***param*** は破棄するキャプチャパラメータへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

（高分解能エリアセンサ制御ライブラリサンプルコード５）を参照してください。

留意事項

ありません

Lib_high_res_set_param

機　能

キャプチャパラメータの設定

形　式

```
#include "f_hres.h"
int Lib_high_res_set_param( void *param );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

カレントチャネルのキャプチャパラメータを設定します。

① ***param*** は設定するキャプチャパラメータへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

（高分解能エリアセンサ制御ライブラリサンプルコード５）を参照してください。

留意事項

○実装されているキャプチャと、キャプチャパラメータの対応キャプチャが一致しない場合はエラーとなります。

Lib_high_res_get_param

機　能

キャプチャパラメータの取得

形　式

```
#include "f_hres.h"
int Lib_high_res_get_param( void *param );
```

９０１	９０２	９０３	９０４	FVL/LNX
	○			○

解　説

カレントチャネルのキャプチャパラメータを取得します。

① ***param*** は取得するキャプチャパラメータへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

（高分解能エリアセンサ制御ライブラリサンプルコード５）を参照してください。

留意事項

○実装されているキャプチャと、キャプチャパラメータの対応キャプチャが一致しない場合はエラーとなります。

（高分解能エリアセンサ制御ライブラリサンプルコード１）

ＦＣ１３００（竹中システム機器製）から画像を取り込みモニターに表示します。

```
#include "f_hres.h"
#include "f_stdlib.h"
#include "f_gui.h"
#include "f_video.h"
#include "f_time.h"

void main( void )
{
  /* カーソル初期化 */
  Lib_init_cursor();
  /* 高分解能エリアセンサ制御ライブラリ初期化 */
  if( NORMAL_RETURN == Lib_high_res_init( FC1300 ) )
  {

        /* 画像取り込み（画像取り込み完了待ちあり） */
        Lib_high_res_freeze( GRAY_PLANE, TRANSMIT );
        Lib_display_keyinput( 0, 0, "確認" );
        /* 高分解能エリアセンサ制御ライブラリ終了 */
        Lib_high_res_finish();
  }
}
```

（高分解能エリアセンサ制御ライブラリサンプルコード２）

ＦＣ１３００（竹中システム機器製）から画像を取り込みモニターに表示します。
（画像取り込み中に別の処理をします。）

```
#include "f_hres.h"
#include "f_stdlib.h"
#include "f_gui.h"
#include "f_video.h"
#include "f_time.h"

void main( void )
{
  int  flag;

  /* カーソル初期化 */
  Lib_init_cursor();
  /* 高分解能エリアセンサ制御ライブラリ初期化 */
  if( NORMAL_RETURN == Lib_high_res_init( FC1300 ) )
  {
        /* 画像取り込み（画像取り込み完了待ちなし） */
        Lib_high_res_xfreeze();

        /** 画像取り込み以外の処理 **/

        /* 画像取り込み完了待ちループ */
        while( 1 )
        {
              /* 画像取り込み状態取得 */
              flag = Lib_high_res_get_video_status();
              if( flag != TRANSMITING )     break;
        }
        if( flag == FINISHED_TRANSMIT )
        {     /* 画像表示 */
              Lib_ex_video_transmit( -1, GRAY_PLANE, EXTRACT_MODE );
              Lib_display_keyinput( 0, 0, “確認” );
        }
        else
        {     /* 画像取り込み強制終了 */
              Lib_high_res_stop();
        }
        /* 高分解能エリアセンサ制御ライブラリ終了 */
        Lib_high_res_finish();
  }
}
```

（高分解能エリアセンサ制御ライブラリサンプルコード３）

キャプチャボードを２枚使用し、２台のＦＣ１３００から画像を取り込みます。

```
#include "f_hres.h"
#include "f_stdlib.h"
#include "f_gui.h"
#include "f_video.h"
#include "f_time.h"

void main( void )
{
   int     flag0, flag1;

   /* カーソル初期化 */
   Lib_init_cursor();
   /* 高分解能エリアセンサ制御ライブラリ初期化 */
   Lib_high_res_set_video_channel( 0 );
   flag0 = Lib_high_res_init( FC1300 );
   Lib_high_res_set_video_channel( 1 );
   flag1 = Lib_high_res_init( FC1300 );
   if( NORMAL_RETURN == flag0 && NORMAL_RETURN == flag1 )
   {

          /* 画像取り込み（画像取り込み完了待ちあり） */
          Lib_high_res_set_video_channel( 0 );
          Lib_high_res_freeze( GRAY_PLANE, TRANSMIT );
          Lib_display_keyinput( 0, 0,“ビデオチャネル０”);
          Lib_high_res_set_video_channel( 1 );
          Lib_high_res_freeze( GRAY_PLANE, TRANSMIT );
          Lib_display_keyinput( 0, 0,“ビデオチャネル１”);
   }
   /* 高分解能エリアセンサ制御ライブラリ終了 */
   if( NORMAL_RETURN == flag0 )
   {
          Lib_high_res_set_video_channel( 0 );
          Lib_high_res_finish();
   }
   if( NORMAL_RETURN == flag1 )
   {
          Lib_high_res_set_video_channel( 1 );
          Lib_high_res_finish();
   }
}
```

（高分解能エリアセンサ制御ライブラリサンプルコード４）

（例１）FC1300（竹中システム機器製）を内部トリガーモードで使用する場合の例を示します。

```
int hr_init( void )
{
  UWORD   uwImageX, uwImageY, uwDelayX, uwDelayY, uwExt_Trg;
  WORD    wTrigger, wParaMode;
  WORD    re;



  uwImageX = 1280;
  uwImageY = 1030;
  uwDelayX = 128;
  uwDelayY = 1;
  wTrigger = -2;
  uwExt_Trg= 0;
  wParaMode = 0;
  re = Lib_high_res_xinit( uwImageX, uwImageY, uwDelayX,
                                  uwDelayY, wTrigger, uwExt_Trg, wParaMode );

  return( re );
}
```

（例２）FC1300（竹中システム機器製）をノーマルモードで使用する場合の例を示します。

```
int hr_init( void )
{
  UWORD   uwImageX, uwImageY, uwDelayX, uwDelayY, uwExt_Trg;
  WORD    wTrigger, wParaMode;
  WORD    re;



  uwImageX = 1280;
  uwImageY = 1030;
  uwDelayX = 128;
  uwDelayY = 1;
  wTrigger = 0;  /* 例１とはここが違います */
  uwExt_Trg= 0;
  wParaMode = 0;
  re = Lib_high_res_xinit( uwImageX, uwImageY, uwDelayX,
                                  uwDelayY, wTrigger, uwExt_Trg, wParaMode );

  return( re );
}
```

（例３）FC1300（竹中システム機器製）を外部トリガモードで使用する場合の例を示します。

```
int hr_init( void )
{
  UWORD   uwImageX, uwImageY, uwDelayX, uwDelayY, uwExt_Trg;
  WORD    wTrigger, wParaMode;
  WORD    re;



  uwImageX = 1280;
  uwImageY = 1030;
  uwDelayX = 128;
  uwDelayY = 1;
  wTrigger = -2;
  uwExt_Trg= 1;          /* 例１とはここが違います */
  wParaMode = 0;
  re = Lib_high_res_xinit( uwImageX, uwImageY, uwDelayX, uwDelayY,
                                                  wTrigger, uwExt_Trg, wParaMode );

  return( re );
}
```

（例４）FC1500（竹中システム機器製）をノーマルトリガモードと内部トリガモードで使用する場合の例を示します。

```
int hr_init( void )
{
  UWORD   uwImageX, uwImageY, uwDelayX, uwDelayY, uwExt_Trg;
  WORD    wTrigger, wParaMode;
  WORD    re;

  /* 高分解能エリアセンサ制御ライブラリ初期化 */
  uwImageX = 1376;       /* X [1392-(1392%32)] */
  uwImageY = 1040;
  uwDelayX = 192;
  uwDelayY = 8;
  wTrigger = 0;
  uwExt_Trg= 0;
  wParaMode = BIT8_MODE;         /* 8bit */
  re = Lib_high_res_xinit( uwImageX, uwImageY, uwDelayX,
                                  uwDelayY, wTrigger, uwExt_Trg, wParaMode );

  return( re );
}
```

（例５）FC1500 を外部トリガモードで使用する場合の例を示します。
カメラ側の設定で『トリガパルス幅指定』にした場合は、
下記のトリガパルス幅(wTrigger)の期間だけ露光します。
『高速シャッタ』で使用する場合は、カメラ側のシャッタ速度に依存します。
何れもトリガパルス幅(wTrigger)には０以外を指定する必要があります。

```
int hr_init( void )
{
  UWORD   uwImageX, uwImageY, uwDelayX, uwDelayY, uwExt_Trg;
  WORD    wTrigger, wParaMode;
  WORD    re;

  /* 高分解能エリアセンサ制御ライブラリ初期化 */
  uwImageX = 1376;       /* X [1392-(1392%32)] */
  uwImageY = 1040;
  uwDelayX = 192;
  uwDelayY = 8;
  wTrigger = -1023;      /* 1023H=1/15.625 秒 */
  uwExt_Trg= 1;          /* 外部トリガ */
  wParaMode = BIT8_MODE;         /* 8bit */
  re = Lib_high_res_xinit( uwImageX, uwImageY, uwDelayX,
                                uwDelayY, wTrigger, uwExt_Trg, wParaMode );

  return( re );
}
```

（例６）FC1500 を FC1300 相当の分解能で使用する場合の例を示します。
CCD サイズは同じですが分解能が異なり、使用する映像の範囲が
狭くなるためそのまま置き換えると拡大したように見えます。

```
int hr_init( void )
{
  UWORD   uwImageX, uwImageY, uwDelayX, uwDelayY, uwExt_Trg;
  WORD    wTrigger, wParaMode;
  WORD    re;

  /* 高分解能エリアセンサ制御ライブラリ初期化 */
  uwImageX = 1280;       /* X [1392-(1392%32)] */
  uwImageY = 1030;
  uwDelayX = 240;
  uwDelayY = 13;
  wTrigger = 0;
  uwExt_Trg= 0;          /* ノーマルモード */
  wParaMode = BIT8_MODE;         /* 8bit */
  re = Lib_high_res_xinit( uwImageX, uwImageY, uwDelayX,
                                uwDelayY, wTrigger, uwExt_Trg, wParaMode );

  return( re );
}
```

（高分解能エリアセンサ制御ライブラリサンプルコード５）

カメラコードにFC1500Fを指定して、サンプリングクロックの位相のみを反転に変更する例を示します。
カメラコードにFC1500Fを指定して、サンプリングクロックの位相のみを反転に変更する例を示します。

（例１）FC1300（竹中システム機器製）を内部トリガーモードで使用する場合の例を示します。

```
int hr_set_reverse( void )
{
  CAPTPRM_AREACAM_FHC3310_T  *tpAreaCam1;
  int  re;

  re = ERROR_RETURN;
  tpAreaCam1=(CAPTPRM_AREACAM_FHC3310_T*)Lib_high_res_create_param
                                                ( CAPTPRM_AREACAM_FHC3310_ID );
  if( NULL != tpAreaCam1 )
  {
    tpAreaCam1->iCamCode= FC1500F;
    tpAreaCam1->uiPhase = 1;     /* 位相反転 */
    re = Lib_high_res_set_param( tpAreaCam1 );

    Lib_high_res_destroy_param( tpAreaCam1 );
  }

  return( re );
}
```

6.3　サンプルソフトウェア説明

本ライブラリには以下のサンプルソフトウェアが付属しています。

①ＦＣ１３００用通信ソフト
　　ソース　　FC13_T.C
　　実行ファイル　　FC13_T.EXE

②高分解能エリアカメラ用画像取り込みソフト

キャプチャ	ソース	実行ファイル
IPM-8500D	HR331LV.ｃ	H_RES_T.exe
FHC331(LV),FHC3321	HR331LV.ｃ	FHC331LV.exe
FHC3310(A)	HR3310.ｃ	HR3310.exe

ソースファイルは、csc902¥sample に
実行ファイルは、csc902¥exe にそれぞれインストールされています。

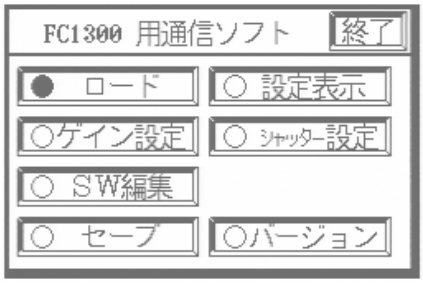
次ページに、それぞれのソフトウェアの概略と操作方法を示します。
本ライブラリを使用し開発を行う場合の参考になさってください。

①ＦＣ１３００用通信ソフト

本ソフトウェアはＦＣ１３００のシリアルコマンドを利用して、カメラの設定を行えるようにしたものです(ＦＣ１３２０Ｖは通信によるカメラ設定機能を搭載していないため使用できません)。
カメラの設定の詳細はＦＣ１３００取扱説明書を参照してください。
また、現在の設定の確認、保存等も容易に行えます。

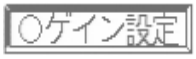
サンプルソフトを実行すると初期メニューが表示されます。

（初期メニュー）

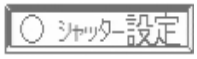


：Ａ～Ｆの中から選択されたページの設定データを読み込みます。

：現在の設定を表示します。

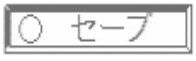


：ゲイン設定モードに移ります。



：シャッター設定モードに移ります。

：シャッタースイッチの設定を変更します。



：Ａ～Ｆの中から選択されたページに現在の設定データを書き込みます。

：本ソフトウェアのバージョンを表示します。

［注］本ソフトウェアはＦＣ１３００と通信できる状態でないと使用できません。
メニューが表示されない場合、またはエラーメッセージが表示された場合はＦＣ１３００との接続を確認のうえ、本ソフトウェアを再起動してください。

■ロード

初期メニューでロードを選択するとページ選択メニューが表示されます。

（ページ設定メニュー）

本メニューで選択されたページの設定データが現在の設定となります。

■ゲイン設定

初期メニューでゲイン設定を選択するとゲイン設定メニューが表示されます。

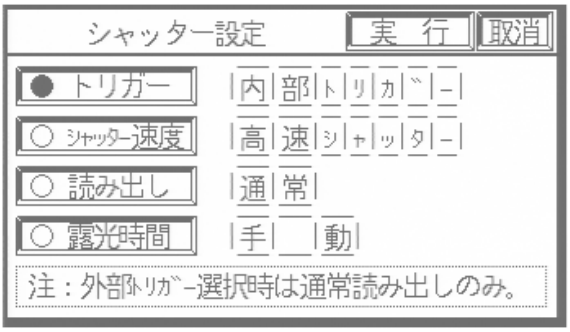
（ゲイン設定メニュー）

本メニューで各ゲインの値を変更できます。

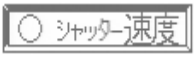
■シャッター設定

初期メニューでシャッター設定を選択するとシャッター設定メニューが表示されます。

（シャッター設定メニュー）



：外部トリガーまたは内部トリガーを選択することができます。



：高速シャッターまたは低速シャッターを選択することができます。

：通常読み出しまたは倍速読み出しを選択することができます。

：既定値（各スイッチに設定されている値で設定するモードです。）
手動（任意の値を入力するモードです。）

［注］ＦＣ１３００で倍速読み出しを使用するには、トリガーを内部トリガーにして使用してください。

■ＳＷ編集

初期メニューでＳＷ編集を選択するとＳＷ編集メニューが表示されます。

ＳＷ編集では各スイッチに設定されている露光時間を変更することができます。

：高速シャッター時の各設定時間を変更します。

：低速シャッター時の各設定時間を変更します。

■セーブ

初期メニューでセーブを選択するとページ選択メニューが表示されます。

本メニューで選択されたページに現在の設定データを記録します。
［注］記録はカメラの設定用ページに記録されます。

■バージョン

本ソフトウェアのバージョンが表示されます。

②高分解能エリアカメラ用画像取り込みソフト

サンプルソースをコンパイル、リンクすると実行可能なサンプルプログラムとなります。
・IPM-8500D,FHC331,FHC331LV,FHC3321. . . . . HR331LV.c
・FHC3310(A). HR3310.c
サンプルソースは、csc902¥sampleディレクトリにインストールされています。
以下はシステムの言語設定が日本語の場合の操作説明です。

本ソフトウェアでは、高分解能エリアカメラライブラリで対応しているカメラの映像を白黒モニター上に表示することができます。

ソフトを起動するとカメラ選択メニューが表示されます。

(カメラ選択メニュー)

使用するキャプチャボードにより、選択可能なカメラ・モードが列挙されます。
接続されているカメラを選択してください。

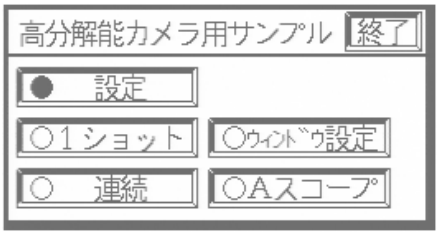
また、カメラ選択メニュー１上で

キーを押すとソフトを終了させることができます。

注：本ソフトウェアを起動する前にカメラ側の設定をしておいてください。

カメラの選択が終了すると初期メニューが表示されます。

（初期メニュー）



：画像取り込みプレーン、モニターへの出力、画像サイズの設定を行います。

：キーを押したタイミングで画像を取り込みます。

：等倍画像の出力ウィンドウの調整をします。

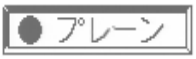
：キー入力があるまで連続で画像を取り込みます。

：取り込んだ画像をＡスコープ表示します。

■設定

初期メニューで設定を選択すると設定メニューが表示されます。

（設定メニュー）



：画像取り込みプレーンをグレイ、２値、両方の中から選択できます。

：スイッチをＯＮにすると、画像を白黒モニターに表示します。

：白黒モニターへの表示を等倍、縮小のいずれかに選択できます。

■ウィンドウ設定

基本ライブラリ Lib_extraction_display_memory の項をお読みください。

■Ａスコープ

基本ライブラリ Lib_ex_ascope の項をお読みください。

7. 高分解能モード移行ライブラリ

本項では当社のライブラリとＩＰＭ－８５００Ａ／ＩＰＭ－８５００Ｄのライブラリを組み合わせ、高分解能センサ画像入力ルーチンを独自に作成するための説明が記載されています。

当社の高分解能関係ライブラリにおいて、サポート外の高分解能センサを使用する際にご利用ください。

Lib_ExtraCameraInit

機　能

高分解能モードの初期化

形　式

```
#include "f_video.h"
int Lib_ExtraCameraInit( int hlength, int vlength );
```

９０１	９０２	９０３	９０４
	○		

解　説

ＲＳ－１７０画像入力から、高分解能センサ画像入力へ切替えるための初期化処理を行います。

① ***hlength*** は水平方向画像サイズです。

② ***vlength*** は垂直方向画像サイズです。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

例

7.1　処理の流れ以降の説明をお読みください。

留意事項

○高分解能センサ画像を入力する場合は必ずコールしておいてください。
○本ライブラリを呼び出し後、画像メモリサイズは、指定したサイズになります。

Lib_ExtraCameraFinish

機　能

高分解能モードの終了

形　式

```
#include "f_video.h"
int Lib_ExtraCameraFinish( void );
```

９０１	９０２	９０３	９０４
	○		

解　説

高分解能センサ画像入力から、ＲＳ－１７０画像入力へ切替えるための処理を行います。

戻り値

処理結果

値	定　　数	意　　味
０	NORMAL_RETURN	正常終了しました。
－１	ERROR_RETURN	異常終了しました。

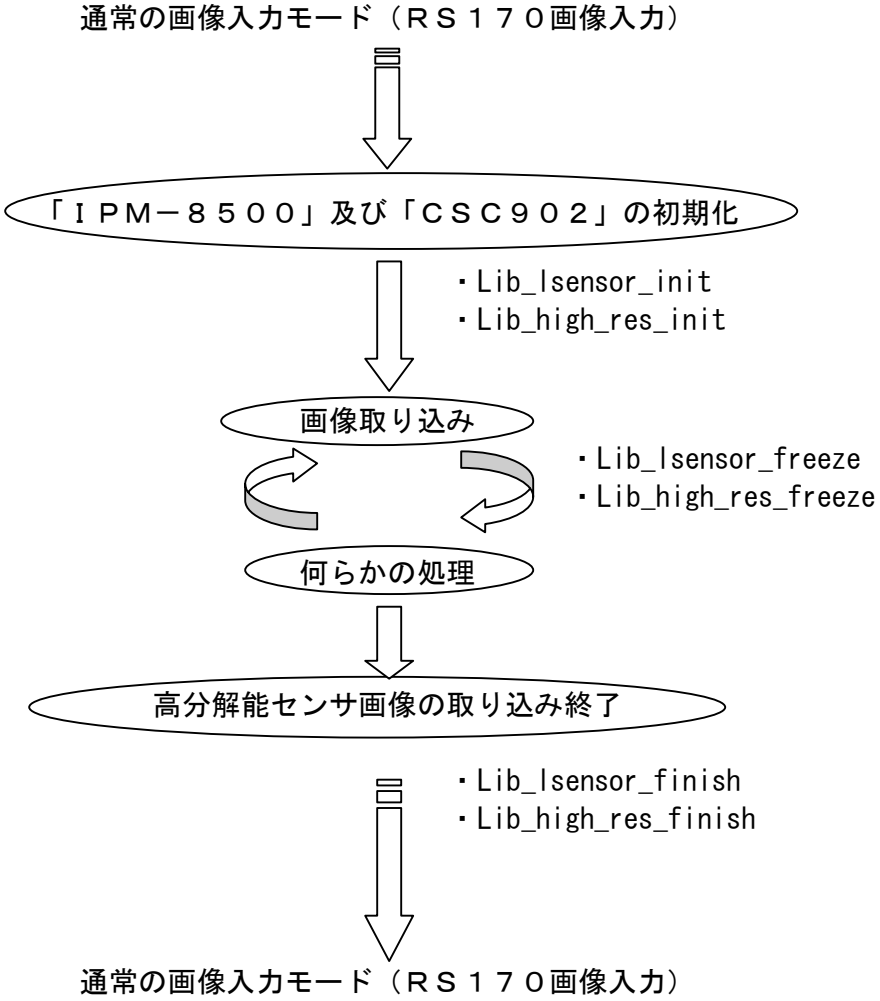
例

7.1　処理の流れ以降の説明をお読みください。

留意事項

○高分解能センサ画像入力を終了する場合は必ずコールしてください。
○本ライブラリを呼び出し後、画像メモリサイズは、５１２Ｘ４８０になります。

7.1　処理の流れ



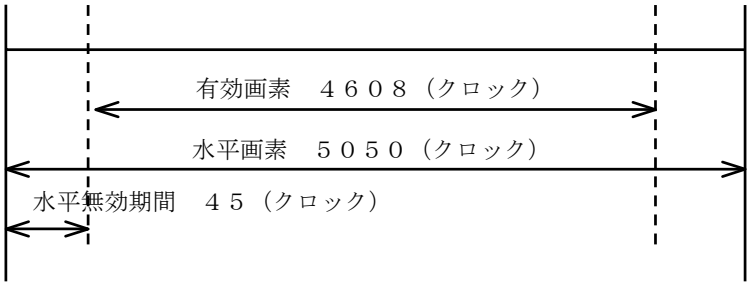
上図に高分解能センサを使用し画像入力を行い、入力画像の処理を行うための一例を示しました。
また、参考までにそれぞれの処理に対応するライブラリを、各処理の右下部分に記載してあります。
（上段がＩＰＭ－８５００Ａ使用時、下段がＩＰＭ－８５００Ｄ使用時）
実際に画像入力ライブラリを作成される際には、上図に示した３つのライブラリと同様の機能を持つライブラリを作成していただくことになります。
次ページから各処理部分のＣ言語サンプルコードと簡単な説明を記載させていただきます。
ライブラリを作成される際の参考になさってください。

7.2　「IPM-8500」及び「902」の初期化

最初に、高分解能センサを使用するためＩＰＭ－８５００Ａ／ＩＰＭ－８５００Ｄの認識、カメラパラメータの設定、９０２側の画像入力モード切替えなどを行います。

次ページにＩＰＭ－８５００Ａを使用する場合の例として

「カメラ条件」
水平期間　　：５０５０クロック
水平有効期間　：５０００クロック
水平無効期間　：　　４５クロック
水平有効期間中の有効画素　：５０００クロックのうち水平有効開始から、
４６０７クロックまでの４６０８クロックが有効画素。



１フレームあたりの垂直期間：４３２０水平ライン
（注：垂直期間はカメラによらず任意に設定できます。）

カメラのピクセルクロック：１０ＭＨｚ

カメラのピクセルクロックと出力データの関係：１ピクセルクロックあたり８Ｂｉｔ、
１ピクセルのデータを出力。

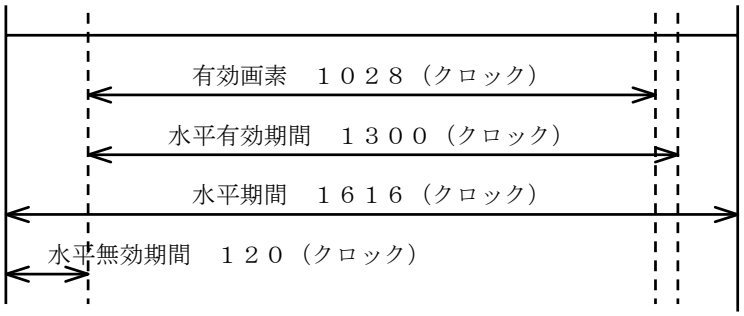
のラインセンサカメラに対する初期化部分のＣ言語コードを示します。

（初期化部分　Ｃ言語サンプルコード：ＩＰＭ－８５００Ａ用）

```
#include <stdio.h>
#include "gbllib.h"
#include "f_video.h"

int init_85A( void );

int init_85A( void )
{
  IPM85_SYS  tMap;
  int     re;

  /* ＩＰＭ－８５００Ａの初期化 */
  re = -4;
  IPM_sys_init();                                  /* システムの初期化 */
  if( 0 < IPM_search( &tMap, 1 ) )                 /* ボードの検索 */
  {
        re = -3;
        if( 0 == IPM_bdset( &tMap, 0 ) )           /* ボードの割り当て */
        {
              if( 0 == IPM_bd_init( 0 ) )          /* ボードの初期化 */
                    re = 0;
        }
  }

  /* ＩＰＭ－８５００Ａに対するカメラ設定の入力 */
  if( 0 == re )
  {
        IPM_set_sysc( 0, 2 );                      /* クロック設定 */
        IPM_set_start( 0, 45 );                    /* 水平方向開始ブランキング */
        IPM_set_lenb( 0, 4608 );                   /* 水平方向取り込み期間 */
        IPM_set_end( 0, 700 );                     /* 水平有効期間調整値 */
        IPM_set_line_count( 0, 4320 );             /* 垂直方向ライン数 */
        IPM_set_camera_rate( 0, 0 );               /* クロック対ビデオ比 */
  }

  /* ＣＳＣ９０２システム拡張画像メモリへ変更 */
  if ( 0 == re )
  {
        Lib_freeze( NOT_TRANSMIT );                /* 画像凍結 */

        re = -2;
        if( 0 == Lib_ExtraCameraInit( 4608, 4320 ) )
              re = 0;
  }

  return( re );
}
```

続いてＩＰＭ－８５００Ｄ使用時の例として

「カメラ条件」
水平期間　　：１６１６クロック
水平有効期間　　：１３００クロック
水平無効期間　　：　１２０クロック
水平有効期間中の有効画素：１３００クロックのうち水平有効開始から、
１２７９クロック目の１２８０クロックが有効画素。



１フレームあたりの垂直期間：１０３０水平ライン

カメラのピクセルクロックと出力データの関係：１ピクセルクロックあたり８Ｂｉｔ、
１ピクセルのデータを出力。

シャッタータイミング：ボードからのトリガー（負論理）で使用。

のエリアセンサカメラに対する初期化部分のＣ言語コードを示します。

（初期化部分　Ｃ言語サンプルコード：ＩＰＭ－８５００Ｄ用）

```
#include <stdio.h>
#include "gbllib.h"
#include "f_video.h"

int init_85D( void );

int init_85D( void )
{
  IPM85_SYS  tMap;
  int    re;

  /* ＩＰＭ－８５００Ｄの初期化 */
  re = -4;
  IPM_sys_init();                              /* システムの初期化 */
  if( 0 < IPM_search( &tMap, 1 ) )             /* ボードの検索 */
  {
        re = -3;
        if( 0 == IPM_bdset( &tMap, 0 ) )       /* ボードの割り当て */
        {
              if( 0 == IPM_bd_init( 0 ) )      /* ボードの初期化 */
                    re = 0;
        }
  }

  /* ＩＰＭ－８５００Ｄに対するカメラ設定の入力 */
  if( 0 == re )
  {
        IPM_set_camera_rate( 0, 4 );           /* クロック外部入力有効 */
        IPM_set_exline( 0, 1 );                /* 外部ライン信号有効 */
        IPM_set_exframe( 0, 1 );               /* 外部フレーム信号有効 */
        IPM_set_start( 0, 120 );               /* 水平方向開始ブランキング */
        IPM_set_lenb( 0, 1280 );               /* 水平方向取り込み期間 */
        IPM_set_end( 0, 120 );                 /* 水平有効期間調整値 */
        IPM_set_line_count( 0, 1 );            /* 取り込みフレーム枚数 */
        IPM_set_exppor( 0, 1 );                /* トリガー（負論理）設定 */
        IPM_set_expcnt( 0, 2000 );             /* トリガーパルス長設定 */
  }
  /* ＣＳＣ９０２システム拡張画像メモリへ変更 */
  if ( 0 == re )
  {
        Lib_freeze( NOT_TRANSMIT );            /* 画像凍結 */
        re = -2;
        if( 0 == Lib_ExtraCameraInit( 1280, 1030 ) )
              re = 0;
  }
  return( re );
}
```

7.3　画像取り込み

次に、前ページで設定したカメラに対して画像取り込み指示を出す場合のＣ言語ソースコードを示します。

（画像取り込み　Ｃ言語サンプルコード：ＩＰＭ－８５００Ａ用）

```
#include <stdio.h>
#include "gbllib.h"
#include "f_stdlib.h"
#include "f_pinf.h"
#include "f_video.h"
#include "f_gray.h"

int freeze_85A( void );

int freeze_85A( void )
{
  char    *bpAdr;
  int     wSize;
  int     re;

  re = -1;
  /* ＣＳＣ９０２カレントグレイメモリーアドレス取得 */
  bpAdr = Lib_adrs_gray_memory( -1 );
  if( (BYTE *)NULL != bpAdr )
  {
        wSize = Lib_get_fx_size()* Lib_get_fy_size();

        /* 画像格納アドレスの設定 */
        IPM_set_MEMADR( 0, bpAdr, wSize );
        /* 画像取り込み開始（アイドル状態） */
        IPM_gray_GRAB( 0 );

        /* 転送待ち */
        while( 1 )
        {
              if( -2 != IPM_GRAB_STS( 0 ) )
              break;
        }
        /* フレームバッファへの転送 */
        Lib_ex_video_transmit( -1, GRAY_PLANE, EXTRACT_MODE );
        re = 0;
  }
  return( re );
}
```

（画像取り込み　Ｃ言語サンプルコード：ＩＰＭ－８５００Ｄ用）

```
#include <stdio.h>
#include "gbllib.h"
#include "f_stdlib.h"
#include "f_pinf.h"
#include "f_video.h"
#include "f_gray.h"

int freeze_85D( void );

int freeze_85D( void )
{
  char    *bpAdr;
  int     wSize;
  int     re;

  re = -1;
  /* ＣＳＣ９０２カレントグレイメモリーアドレス取得 */
  bpAdr = Lib_adrs_gray_memory( -1 );
  if( (BYTE *)NULL != bpAdr )
  {
        wSize = Lib_get_fx_size()* Lib_get_fy_size();

        /* 画像格納アドレスの設定 */
        IPM_set_MEMADR( 0, bpAdr, wSize );
        /* 画像取り込み開始（アイドル状態） */
        IPM_gray_GRAB( 0 );
        /* トリガー信号発生 */
        IPM_expose( 0 );

        /* 転送待ち */
        while( 1 )
        {
              if( -2 != IPM_GRAB_STS( 0 ) )
              break;
        }
        /* フレームバッファへの転送 */
        Lib_ex_video_transmit( -1, GRAY_PLANE, EXTRACT_MODE );
        re = 0;
  }
  return( re );
}
```

7.4　高分解能センサ画像の取り込み終了

最後に、高分解能センサの使用を終了し通常のカメラ入力に戻る場合のＣ言語ソースコードを示します。

（終了部分　Ｃ言語サンプルコード：ＩＰＭ－８５００Ａ用）

```
#include <stdio.h>
#include "gbllib.h"
#include "f_stdlib.h"
#include "f_pinf.h"
#include "f_video.h"
#include "f_gray.h"

int finish_85A( void );

int finish_85A( void )
{
  int  re;

  IPM_sys_free( 0 );

  re = Lib_ExtraCameraFinish();
  /* 画像凍結解除 */
  Lib_freerun();

  return( re );
}
```

（終了部分　Ｃ言語サンプルコード：ＩＰＭ－８５００Ｄ用）

```
#include <stdio.h>
#include "gbllib.h"
#include "f_stdlib.h"
#include "f_pinf.h"
#include "f_video.h"
#include "f_gray.h"

int finish_85D( void );

int finish_85D( void )
{
  int  re;

  IPM_sys_free( 0 );

  re = Lib_ExtraCameraFinish();
  /* 画像凍結解除 */
  Lib_freerun();

  return( re );
}
```

これで高分解能センサ制御ライブラリの説明は終了です。
ＩＰＭ－８５００Ａ／ＩＰＭ－８５００Ｄ制御ライブラリの詳しい説明は、
「ＩＰＭ８５００Ａ／ＩＰＭ８５００Ｄ　ＤＯＳ版ドライバ　ライブラリリファレンス」をお読みください。

索　引

E

Lib_ExtraCameraFinish 159
Lib_ExtraCameraInit.....158

H

Lib_high_res_convert_for_bit16b.....131
Lib_high_res_create_param.....133
Lib_high_res_destroy_param.....138
Lib_high_res_finish.....121
Lib_high_res_freeze 122
Lib_high_res_get_param.....140
Lib_high_res_get_video_channel.....127
Lib_high_res_get_video_channel_num.....130
Lib_high_res_get_video_status.....124
Lib_high_res_init.....112
Lib_high_res_set_param 139
Lib_high_res_set_video_channel 126
Lib_high_res_SetGrabCondition 128
Lib_high_res_stop 125
Lib_high_res_xfreeze.....123
Lib_high_res_xinit.....116

L

Lib_lsensor_adjust_bin_level 71
Lib_lsensor_adrs_bin_memory..... 52
Lib_lsensor_adrs_gray_memory..... 50
Lib_lsensor_alloc_bin_memory 48
Lib_lsensor_alloc_gray_memory 46
Lib_lsensor_bin_memory_cls..... 63
Lib_lsensor_change_bin_memory 62
Lib_lsensor_change_gray_memory 61
Lib_lsensor_change_memory 60
Lib_lsensor_continuous_grab..... 84
Lib_lsensor_create_param 87
Lib_lsensor_destroy_param 91
Lib_lsensor_extraction_display_memory..... 69
Lib_lsensor_finish..... 32
Lib_lsensor_free_bin_memory..... 56
Lib_lsensor_free_gray_memory 54
Lib_lsensor_freeze 38
Lib_lsensor_get_bin_memory..... 59
Lib_lsensor_get_error_inf..... 45
Lib_lsensor_get_extract_window 66
Lib_lsensor_get_grab_count..... 85
Lib_lsensor_get_gray_memory..... 58
Lib_lsensor_get_line_count 86
Lib_lsensor_get_param..... 93
Lib_lsensor_get_video_channel..... 43
Lib_lsensor_get_video_channel_num 44
Lib_lsensor_get_video_status..... 37
Lib_lsensor_gray_memory_cls 64
Lib_lsensor_hardver 82
Lib_lsensor_init 27
Lib_lsensor_memory_reset..... 73
Lib_lsensor_set_extract_window 67
Lib_lsensor_set_param..... 92
Lib_lsensor_set_video_channel 41
Lib_lsensor_SetEncoder 79
Lib_lsensor_SetGrabCondition 77
Lib_lsensor_SetGrabMode 74
Lib_lsensor_test_input 39
Lib_lsensor_video_input..... 33
Lib_lsensor_video_input_abort 83
Lib_lsensor_video_transmit 65
Lib_lsensor_xvideo_input..... 35

902 **高分解能センサライブラリ説明書**

２００５年１月第１３版第１刷発行

発行所　　株式会社ファースト

本　　社　〒242-0001　神奈川県大和市下鶴間２７９１－５

ユーザー・サポート　　FAX　046-272-8692　　TEL　046-272-8691
E-mail : support@fast-corp.co.jp

B-000696